AFRL-ML-WP-TR-2001-4089

SIMULATION ASSESSMENT VALIDATION ENVIRONMENT (SAVE)

Software User's Manual

Paul Cole
Bob Bassett
Marcia Herndon
Paul Collins
Kathy Jacobson
James Poindexter

Lockheed Martin Tactical Aircraft Systems
1 Lockheed Blvd.
Fort Worth, TX 76108

SEPTEMBER 2000

FINAL REPORT FOR PERIOD 01 APRIL 1995 - 30 SEPTEMBER 2000

Approved for public release; distribution is unlimited

MATERIALS AND MANUFACTURING DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7750

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
00092000	N/A	-

Title and Subtitle Simulation Assessment Validation Environment (SAVE) Software Users Manual	**Contract Number** **Grant Number** **Program Element Number**
Author(s) Cole, Paul; Bassett, Bob; Herndon, Marcia; Poindexter, James	**Project Number** **Task Number** **Work Unit Number**
Performing Organization Name(s) and Address(es) Lockheed Martin Tactical Aircraft Systems 1 Lockheed Blvd. Fort Worth, TX 76108	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es) Materials and Manufacturing Directorate Air Force Research Laboratory Air Force Materiel Command Wright-Patterson AFB, OH 45433-7750	**Sponsor/Monitor's Acronym(s)** **Sponsor/Monitor's Report Number(s)**
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes The original document contains color images.	
Abstract	
Subject Terms	
Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 428	

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASC/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

JAMES W. POINDEXTER
Project Engineer
Information Integration Team
Advanced Manufacturing Enterprise Branch

KENNETH A. FEESER
Leader
Information Integration Team
Advanced Manufacturing Enterprise Branch

BRENCH BODEN
Advanced Manufacturing Enterprise Branch
Manufacturing Technology Division

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** September 2000	**3. REPORT TYPE AND DATES COVERED** Final 4/1/1995 - 9/30/2000

4. TITLE AND SUBTITLE
Simulation Assessment Validation Environment (SAVE)
Software User's Manual

5. FUNDING NUMBERS
C F33615-95-C-5538
PE 63800F
PR 2025
TA 50
WU 03

6. AUTHOR(S)
Paul Cole, Bob Bassett, Marcia Herndon, Paul Collins, Kathy Jacobson - Lockheed Martin Tactical Aircraft Systems
James Poindexter - U.S. Air Force, AFRL/MLMS

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Lockheed Martin Tactical Aircraft Systems
1 Lockheed Blvd.
Fort Worth, TX 76108

8. PERFORMING ORGANIZATION REPORT NUMBER

JRES01046-1

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Materials and Manufacturing Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7750
POC: James W. Poindexter, AFRL/MLMS, (937) 904-4351

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-ML-WP-TR-2001-4089

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 Words)

This document provides appropriate documentation for the user of the SAVE Virtual Manufacturing (VM) Environment. Two primary focus areas include use and implementation. In addition, the SAVE Final Report covers the full period of the SAVE Program, from April 1995 through September 2000, and a detailed description of the SAVE software system is documented in the SAVE Software Product End Item Report.

14. SUBJECT TERMS

Virtual Manufacturing (VM), Joint Advanced Strike Technology (JAST), Joint Strike Fighter (JSF), Lean Implementation.

15. NUMBER OF PAGES
432

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIIFICA-TION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** SAR

TABLE OF CONTENTS

Section **Title** **Page**

TABLE OF CONTENTS (Continued)

Section **Title** **Page**

Chapter 2 (continued)

TABLE OF CONTENTS (Continued)

Section **Title** **Page**

Chapter 3

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

Section **Title** **Page**

Chapter 4 (continued)

Chapter 5

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

FOREWORD

Commercial industry is leading the way in implementing the use of modeling and simulation tools to reduce product cost, time to market, etc. The use of these tools results in leaner systems that are more competitive in the global market. The emphasis within commercial industry is not only to stay in business, but become more profit conscious. Many companies are seeing declining revenues and higher profits. How is this possible? They have found ways to reduce cost, in other words make their products more affordable. This occurs in many ways from streamlined production, to the rapid introduction of new products, to strategic partnering, including outsourcing or co-sourcing.

The ability to simulate manufacturing operations, prior to actual production, is having a significant impact on product and process design decision making. Commercial simulation tools have matured rapidly in recent years, but their use is still somewhat limited by the lack of integration among sets of tools to evaluate cost, schedule, and risk. The SAVE program was initiated to address this required integration.

The concept of affordability is a central theme in the Joint Strike Fighter (JSF) program. This is seen in the genesis of the program: combining three products into one to leverage affordability by streamlining development and production cost. In concept, all three derivative aircraft are designed and manufactured jointly, with the exception of parts that are affected by customer specific requirements (e.g., Navy, carrier based models, require additional structural enhancements for the undercarriage). The design and manufacturing effort is characterized by a single effort that encompasses all common parts with a split near the end to handle customer specific requirements. The net result will be an affordable fighter through the leveraging of common design and manufacturing efforts.

This concept was further expanded within the Manufacturing and Producibility Integrated Product and Process Team (IPPT) through the sponsorship of six key initiatives including:

- JSF Manufacturing Capabilities Tool Set (JMCATS) - Developing a tool set for analyzing manufacturing risk and process capabilities with traceability back to basic product requirements and functions.

- JSF Manufacturing Demonstration Program (JMD) - Developing an IPPT process with supporting tools to assess manufacturing cost directly from CAD data bases and to collect manufacturing information needed to drive cost engines.

- Virtual Manufacturing Fast Track Program - An initial JSF demonstration showing the usefulness of virtual manufacturing using an integrated environment of available software design and manufacturing tools.

- Ribbonized, Organized, Integrated (ROI) Wiring Program - A JSF demonstration showing the potential weight and cost savings using an ROI wiring architecture in a tactical aircraft.

- Manufacturing Affordability Development Program (MADP) - A JSF Government Team survey of twelve companies at seventeen facilities to identify pockets of manufacturing and producibility successes which demonstrated affordability potential for the remainder of the industry.

- Simulation Assessment Validation Environment Program (SAVE) - Developing a method of creating a virtual manufacturing environment through the integration of a set of simulation, modeling and analysis tools.

Combined these programs are estimated to achieve a 12%-20% reduction in life cycle cost through demonstration and implementation of improved processes and tools which reflect manufacturing considerations early in design.

These programs were identified as a result of the 1994 Government Led Lean Forum Workshop. The consensus topics from this workshop were Integrated Design and Cost; Modeling and Simulation; Teaming; Factory Operations; and Design for Quality and Producibility. The results of this workshop have led to the JSF sponsored programs listed above. The SAVE program addresses the consensus topic of Modeling and Simulation.

The SAVE program is the integration of best of breed commercial off the shelf tools that support the generation and analysis of data needed to make affordability based decisions. This leads to an ability to perform cost/performance trade studies, thereby enabling the treatment of cost as an independent variable by making cost clearly quantified as design requirements and decisions are made. The integration is leveraging work from other DoD organizations so that high-end results are attainable much faster than is possible without these capabilities. The end result will be a new set of commercially available capabilities that can support the entire JSF customer, prime, team, supplier and user base. SAVE provides the ammunition to drive affordability at all levels in the program. The SAVE program is estimated to contribute to 1%-2% of the life cycle cost savings listed above.

This report documents the guidelines for the implementation and use of SAVE within the JSF or any other manufacturing program.

PREFACE AND ACKNOWLEDGMENTS

This report provides user-level information on the activities and results developed during the entire life of the program and has been prepared for USAF/AFMC/ASC AFRL/MLMS as CDRL number A012 for contract number F33615-95-C-5538.

The following individuals are acknowledged for their work and dedication in achieving the successful results of the program:

James Poindexter	USAF Program Manager
Paul Cole	Lockheed Martin Program Manager
Bob Bassett	Architecture/Infrastructure IPPT Lead
Marcia Herndon	Tool Integration IPPT Lead
Paul Collins	Demonstration/Technology Transfer IPPT Lead
Kathy Jacobson	Enhanced Design Cost Integration IPPT Lead

LIST OF ACRONYMS AND ABBREVIATIONS

ABC	- Activity Based Costing
ADAM	- Affordable Design And Manufacturing
AFMC	- Air Force Materials Command
AFRL	- Air Force Research Laboratory
AIC	- Artificial Intelligence Center
AIMS	- Agile Infrastructure for Manufacturing Systems
ANSI	- American National Standards Institute
API	- Application Programming Interface
ASC	- Aeronautical Systems Center
Assy	- Assembly
ASURE	- Analysis System for Uncertainty and Risk Estimation
BAFO	- Best and Final Offer
BCL	- Batch Control Language
BOM	- Bill of Material
CA	- Cost Advantage
CAD	- Computer Aided Design
CATIA	- Computer Aided Three dimensional Interactive Application
CAx	- Computer Aided (Anything)
CDA	- Concept Demonstrator Airplane
CDE	- Common Desktop Environment
CDF	- Common Data File
CDP	- Concept Demonstrator Program
CDRL	- Contract Data Requirements List
CE	- Concurrent Engineering
CER	- Cost Estimating Relationship
COM	- Common Object Model
CORBA	- Common Object Request Broker Architecture
COSE	- Common Open System Environment
COTS	- Commercial Off the Shelf
CSA	- Close Support Aircraft

CSC	- Computer Science Corporation
CTC	- Concurrent Technologies Corporation
DARPA	- Defense Advanced Research Projects Agency
DB	- Database
DDI	- Decision Dynamics Incorporated
DEAM	- Design Engineering Analysis Model
DFM	- Design for Manufacturability
DICE	- DARPA Initiative on Concurrent Engineering
DID	- Data Item Description
DLL	- Dynamic Link Library
DMC	- Defense Manufacturing Conference
DNS	- Domain Name Service
DoD	- Department of Defense
DOE	- Design of Experiments
E&MD	- Engineering and Manufacturing Development
EAF	- Estimate Adjustment Factor
EAI	- Engineering Animation, Incorporated
EBOM	- Engineering Bill of Materials
ECRC	- Electronic Commerce Resource Center
EDN	- Electronic Design Notebook
EMD	- Engineering Manufacturing Development
ERGO	- Ergonomics Option
ERP	- Enterprise Resource Planning
FA	- Factor AIM
Fab	- Fabrication
GD&T	- Geometric Dimensioning & Tolerancing
GDT	- Geometric Dimensional Tolerancing
GE	- General Electric
GIFF	- Graphical Interface File Format
GII	- Graphic Interactive Interface
GIT	- Georgia Tech Institute of Technology
GRCI	- General Research Corporation International

GSL	- Graphics Simulation Language
GUI	- Graphical User Interface
H/W	- Hardware
HTML	- Hypertext Markup Language
I/F	- Interface
I/O	- Input/Output
IBAM	- Industrial Base Analysis Model
IBM	- International Business Machines
ID	- Identification
IDL	- Interface Definition Language
IGES	- Initial Graphics Exchange Specification
IGRIP	- Interactive Graphics Robot Instruction Program
IIOP	- Internet Inter-ORB Protocol
IP	- Internet Protocol
IP/PD	- Integrated Product/Process Development
IP/PT	- Integrated Product/Process Team
IPAM	- Industrial Production Analysis Model
IPD	- Integrated Product Development
IPPDB	- Integrated Product Process Database
IPPT	- Integrated Product and Process Team
IPT	- Integrated Product Team
IRS	- Interface Requirements Specification
ISO	- International Standards Organization
JAR	- JAVA Application Resource
JAST	- Joint Advanced Strike Technology
JDK	- JAVA Development Kit
JMCAT	- JSF Manufacturing Capabilities Assessment Tool Set
JMD	- JSF Manufacturing Demonstration
JPO	- Joint Program Office
JSF	- Joint Strike Fighter
KB	- Knowledge Base
KC	- Knowledge Center

LAI	- Lean Aircraft Initiative
LCAM	- Life Cycle Analysis Model
LCC	- Life Cycle Cost
LHS	- Left Hand Side
LM	- Lockheed Martin
LMMS	- Lockheed Martin Missiles and Space
LMSW	- Lockheed Martin Skunk Works
LRIP	- Low Rate Initial Production
MADE	- Manufacturing Automation and Design Engineering
MADP	- Manufacturing Affordability Development Program
MBOM	- Manufacturing Bill of Materials
MDF	- Metadata File
ME	- Manufacturing Engineer
MECE	- Multimedia Environment for Concurrent Engineering
Mfg	- Manufacturing
Mgt	- Management
MLMS	- Material Laboratory Manufacturing Systems
MRP	- Manufacturing Resource Planning
MS	- Microsoft
MTI	- Manufacturing Technology Integration
MTR	- Manufacturing Technology Research
N/C	- Numerical Control
NB	- Net Builder
NC	- Numerical Control
NFS	- Network File System
NIIIP	- National Industrial Information Infrastructure Protocol
NIST	- National Institute of Standards and Technology
ODBC	- Open Data Base Connectivity
OMG	- Object Management Group
OO	- Object Oriented
ORB	- Object Request Broker
OSHA	- Operational Safety and Health Administration

OTF	- Operational Task Force
P&W	- Pratt and Whitney
PC	- Personal Computer
PCA	- Physical Configuration Audit
PCM	- Production Cost Model
PDM	- Product Data Manager
PDR	- Preliminary Design Review
PEO	- Program Engineering Office
PIE	- Probabilistic Inference Engine
POOP	- Plain Old Orbix Protocol
QM	- Query Manager
QUEST	- Queuing Event Simulation Tool
R&D	- Research & Development
RADM	- Rear Admiral
RASSP	- Rapid Prototyping of Application Specific Signal Processors
RDB	- Relational Database
RDE	- RASSP Design Environment
RFP	- Request for Proposal
RHS	- Right Hand Side
ROI	- Return on Investment
ROI	- Ribbonized, Organized, Integrated Wiring
RPC	- Remote Procedure Call
S/W	- Software
SAIC	- Science Applications International Corporation
SAVE	- Simulation Assessment Validation Environment
SBD	- Simulation Based Design
SCRA	- South Carolina Research Authority
SDAI	- Software Data Access Interface
SDE	- SAVE Design Environment
SDM	- SAVE Data Model
SDR	- System Design Review
SGI	- Silicon Graphics Incorporated

SLMC	- Sanders a Lockheed Martin Company
SMC	- Systems Modeling Corporation
SQL	- Structured Query Language
SRR	- Scrap, Rework, and Repair
SS	- Source Selection
STAM	- Strategic Technologies Analysis Model
STL	- Stereolithography
T/BAB	- Technical/Business Advisory Board
TBD	- To Be Determined
TCP/IP	- Transmission Control Protocol/Internet Protocol
TDM	- Technical Data Management
UG	- UniGraphics
USAF	- United States Air Force
USC	- University of Southern California
VM	- Virtual Manufacturing
VP	- Virtual Prototyping
VSA	- Variability Simulation Analysis
WFM	- Work Flow Manager
WL	- Wright Laboratory
WPAFB	- Wright-Patterson Air Force Base

NOTICES

1. Deneb Robotics, Inc.
2. Quest® - Registered trademark of Deneb Robotics, Inc.
3. IGRIP® - Registered trademark of Deneb Robotics, Inc.
4. ERGO® - Registered trademark of Deneb Robotics, Inc.
5. Symix Corporation
6. Factor/AIM® - Registered trademark of Symix Corporation
7. Cognition Corporation
8. Cost Advantage™ - trademark of Cognition Corporation
9. SAIC® - Registered trademark of Science Applications International Corporation
10. CATIA® - Registered trademark of Dassault Systemes
11. Dassault Systemes
12. SGI ™ - Trade Mark of Silicon Graphics, Inc.
13. Microsoft ®- Registered trademark of MicroSoft Corporation
14. Windows® ™ - Registered trademark or Trade Mark of MicroSoft Corporation
15. Windows NT® ™ - Registered trademark or Trade Mark of MicroSoft Corp.
16. JMCATS
17. AIX ™ - Trade Mark of International Business Machines
18. IBM® - Registered trademark of International Business Machines
19. UNIX® - Registered trademark of UNIX System Labs, Inc.
20. Collabora®- Registered trademark of Netscape, Inc.
21. Engineering Animation, Inc.

General Notice: Some of the product names used herein have been used for identification purposes only and may be trademarks or registered trademarks of their respective manufacturers.

ABSTRACT

The Joint Strike Fighter (JSF) Simulation Assessment Validation Environment (SAVE) Program provides the capability to assess the manufacturing impacts of both product and manufacturing process design decisions. By integrating Commercial Off-The-Shelf (COTS) modeling and simulation tools into a *seamless virtual environment*, SAVE allows design teams to develop and verify new affordable aircraft concepts before developing expensive hardware.

The SAVE infrastructure utilizes a Common Object Request Broker Architecture (CORBA) based shared Data Model and Work Flow Manager and a commercial *Electronic Collaborative Design Notebook* to integrate a suite of six commercial manufacturing tools which include schedule, factory, assembly, dimensional variability, cost, and risk simulations. In the future, other tools may be added by developing simple SAVE-compliant CORBA wrappers, and SAVE will be available to extend to other problem domains such as operations and support simulations to assess life-cycle issues.

SAVE expects to achieve significant cost savings for the JSF Program by providing integrated design teams the capability to quickly perform “what-if“ studies and accurately define a product’s cost and risk early in the design process. While the SAVE initiative is vital to achieving the affordability goals of the Joint Strike Fighter, the implementation of SAVE in other design and manufacturing environments has the potential to generate equally impressive cost savings.

This document provides user-appropriate documentation of the SAVE Virtual Manufacturing (VM) Environment. Two primary focus areas include use and implementation. First, the document describes the general concept of SAVE along with guidance for its application and use. Next, it provides guidance for implementing a SAVE system including how to calculate the costs and respective benefits for a given implementation. In addition, a number of appendices are included to provide details on the various components of the system.

Chapter 1

Introduction

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

CHAPTER 1 – TABLE OF CONTENTS

LIST OF FIGURES

1.0 Document Overview

This document, The Software User's Manual, is intended to provide a user's view of information about the Simulation Assessment Validation Environment (SAVE) system. It is divided into a series of chapters and appendices for ease of access to the desired information. The core chapters discuss the concepts for operation of the SAVE environment and the steps necessary for its successful use within an Integrated Product Team (IPT) environment. The appendices provide additional detail that would be valuable to users who are applying the system. The following list provides a summary of topics and their location within the document:

Chapter 1	Introduction
Chapter 2	Concept of Operations
Chapter 3	Usage Guidelines
Chapter 4	Implementation Plan
Chapter 5	User's Guide for SAVE-Developed Software
Appendix A	Sample SAVE Usage Scenarios
Appendix B	Use Cases
Appendix C	Data Dictionary
Appendix D	ASURE Wrapper User's Guide
Appendix E	Cost Advantage Wrapper User's Guide
Appendix F	Factor AIM Wrapper User's Guide
Appendix G	IGRIP and QUEST Wrapper User's Guides
Appendix H	VSA-3D Wrapper User's Guide
Appendix I	CATIA Costlink User's Guide
Appendix J	Cost Model User's Guide
Appendix K	Project 98 Wrapper User's Guide
Appendix L	Work Flow Manager User's Guide
Appendix M	Query Manager User's Guide
Appendix N	SAVE Training Material
Appendix O	SAVE Vendor Tool Input/Output Mapping

2.0 Background - Rapid Growth in the Desire to Apply Virtual Manufacturing

When Concurrent Engineering (CE) burst upon the scene in the mid-1980s, acceptance of its concepts grew rapidly. The central precept of CE is the use of multidisciplinary teams representing all facets of the design and manufacturing processes. Each team focuses on the combined problems of product and process development, and strives to eliminate the "over-the-wall" hand-off of data from one organization to the next. Early adopters of the CE approach demonstrated significant improvements in product cost, quality, and time to market. Early consideration of the manufacturing impacts of design decisions clearly results in identifying better designs, early identification of problems, and reduced scrap, rework, repair, and redesign. Application of CE, often called Integrated Product Development (IPD), is now widespread.

As might be expected, the cultural impediments to implementing CE, particularly in large design teams, can be significant. One issue that arises is the varying levels of detail that different team members can bring to bear on a design in the early phases of development. As design concepts are developed teams must balance the impacts on a range of performance considerations, cost, producibility, schedule and risk. Often, the traditional analysis disciplines (performance, weight, structural strength, etc) can make clear, quantified, strong cases for the impacts in their areas. Producibility, cost, schedule, and risk have tended to be more subjective and based on experience rather than analysis. Difficult design decisions tend to be made in favor of the cleanly quantified issue - it won't perform, it weighs too much, it will break, etc. Serious manufacturing issues become concrete at a later, costly phase of a project - during manufacturing.

Recognition of these shortcomings in CE, fueled by the almost explosive growth in affordable computer power, is leading companies to apply the tools of virtual manufacturing.

3.0 The Need for SAVE

Virtual Manufacturing (VM) is the integrated use of design and production models and simulations to support accurate cost, schedule and risk analysis. These modeling and simulation capabilities allow decision-makers to rapidly and accurately determine production impact of product/process alternatives through integrating actual design and production functions with next generation simulation. The use of simulation software to achieve the objectives of virtual manufacturing has been rapidly increasing throughout industry. The potential for these tools to significantly improve affordability and reduce cycle times is widely accepted, but the potential has not been fully achieved.

Many commercial manufacturing simulation tools with excellent capabilities exist on the market today. Although many of these tools rely on similar types of data, differences in internal storage structures and nomenclature have prevented easy tool to tool data integration. Often, large amounts of data must be reentered, at considerable time and expense, to accommodate these differing formats. Some point-to-point solutions do exist between specific tools, but as the number of tools grows, this integration solution becomes unmanageable, and the benefits from using an integrated tool suite go unrealized.

The Simulation Assessment Validation Environment (SAVE) program, led by Lockheed Martin under contract with the Air Force Research Laboratory (AFRL) with funding from the Joint Strike Fighter Program Office, addressed these limitations by developing and implementing an open architecture environment to integrate manufacturing modeling and simulation tools. SAVE also demonstrated this integrated simulation capability to significantly reduce product life cycle costs.

The initial phase of the program, completed in August 1996, established a core tool suite integrated via the Defense Advanced Research Projects Agency (DARPA) developed Rapid Prototyping of Application Specific Signal Processors (RASSP) architecture. The core tool suite incorporates commercial CAD, factory simulation, assembly simulation, schedule simulation, cost and risk modeling capabilities.

During Phase II, the SAVE team developed a Common Object Request Broker (CORBA) based approach to tool integration which provides a solid foundation for ultimate production use and commercialization of SAVE. The CORBA-based infrastructure now includes the SAVE Common Data Model, a Work Flow Manager, and a Query System for interactive access to the Data Model. In addition, commercially available dimension and tolerance simulation capabilities have been added to the VM environment. An Electronic Collaborative Design Notebook is considered essential to SAVE, and although it is not being developed under the contract, a commercially available web-based product was used for Phase II.

4.0 Objectives of SAVE

In recent years, manufacturing modeling and simulation software has seen increased use throughout industry. Rapid advances in computing hardware and software now allow accurate simulations of complex processes. Computer graphics provide Integrated Product/Process Teams (IPPT) with the means to efficiently understand the results of these simulations and make critical design and manufacturing decisions, without resorting to costly physical prototypes.

Growth in the use of virtual manufacturing tools has only been limited by the costly, manual transfer of data among the set of simulation tools. Typically, a design team will use a 2-D or 3-D CAD package for design. The team will then assess the manufacturing impact of product and process decisions through use of a set of virtual manufacturing tools to assess cost, schedule, and risk. The tool capabilities typically include:

- Process planning
- Dimension and tolerance analysis
- Schedule simulation
- Risk analysis
- Assembly simulation
- Factory simulation
- Ergonomic simulation

- Feature-based costing

These tools use much of the same data as input, but each requires different internal data formats. Manual reformatting and reentry of these data are prohibitively costly. The vision of SAVE is to provide a system for integration of simulation codes into an efficient, easy-to-use capability to rapidly assess the manufacturing impacts (cost, schedule, and risk) of product and process design decisions.

A technical solution to the vision of SAVE has been successfully developed, and is nearing commercialization.

5.0 Overview of SAVE Technical Approach

To understand the use of the SAVE Virtual Manufacturing Environment, it is necessary to first gain an understanding of the basics of the technical approach to creating the environment. The two primary elements of SAVE include the simulation tool integration and the tool execution and management infrastructure. The integration allows tools to share common data without concern for their computer platform, the location of that computer, or the language used to program the tool. The execution and management infrastructure facilitates communication, management, and access among the IPPT members using the system.

The components of the SAVE environment and their interfaces within the system are shown in Figure 1-1. Together, these components provide a Virtual Manufacturing capability.

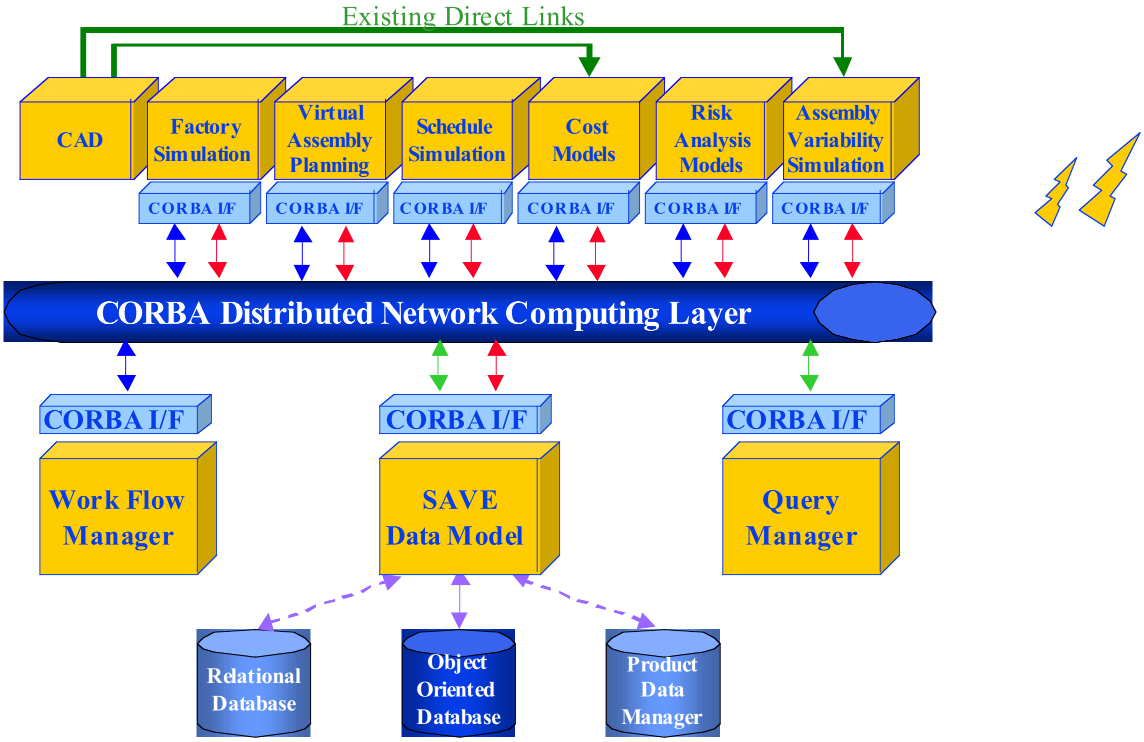


Figure 1-1: SAVE Technical Approach

The concept starts with classes of tools that are generally available in the commercial market and is not dependent on a specific set of vendor tools. These tool classes are identified across the top of Figure 1-1. They include the following:

- CAD
- Factory Simulation
- Virtual Assembly Planning
- Schedule Simulation
- Cost Modeling
- Risk Analysis
- Assembly Variability Simulation

At the heart of the infrastructure is the SAVE Data Model (SDM). It represents the data that is common, or shared, among the tools and provides the contract for data exchange among the various software tools. A graphical overview of the elements of the SDM is shown in Figure 1-2. The definitions and makeup of the data in the SDM are described in detail in Appendix C. The SDM is designed so that it can be implemented with connections to various data storage devices. This capability allows production implementations of SAVE to access the data wherever it resides within the enterprise. For example, tooling data may reside in the PDM System, while material data may be maintained in a separate relational database.

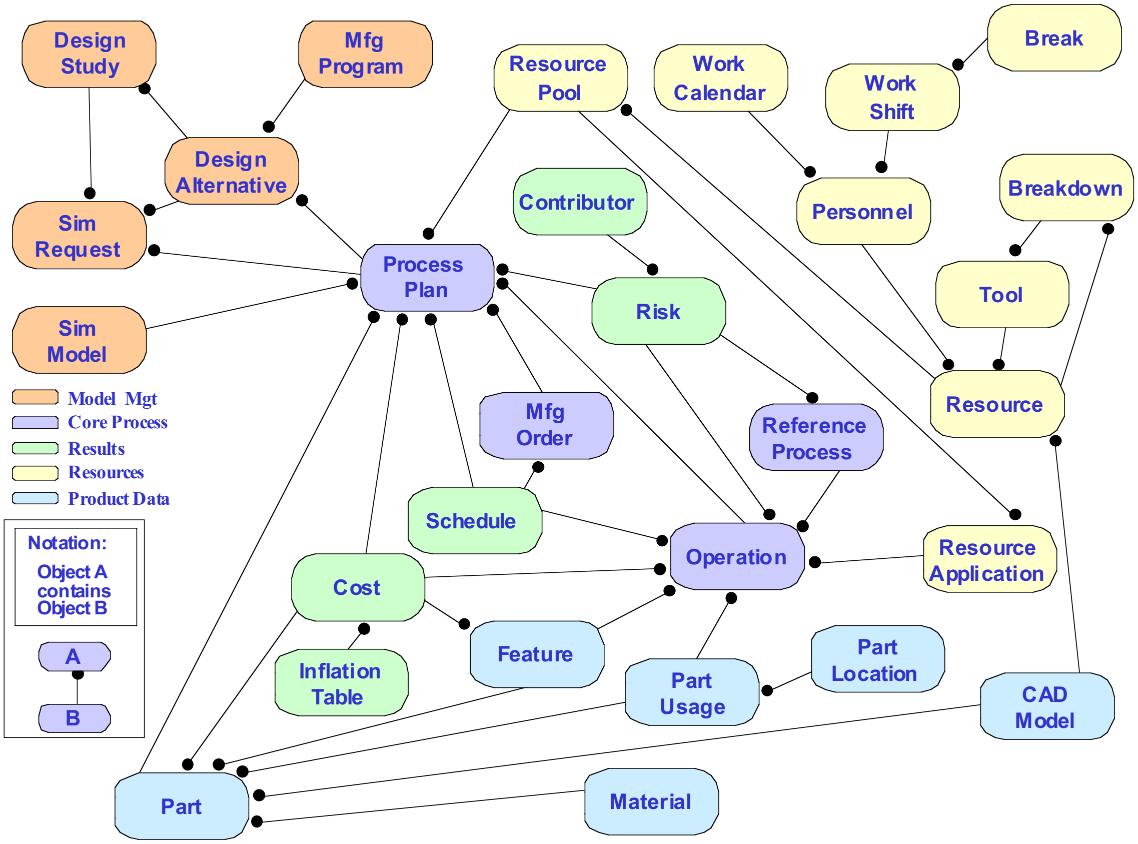


Figure 1-2: The SAVE Data Model

The SAVE architecture also contains a Work Flow Manager (WFM) that provides graphical process modeling and tool execution. More details in the use of the WFM are provided in Appendix L. In order to provide visibility into the information contained within the SDM, a Query Manager (QM) application was developed by the SAVE team. This application, described in Appendix M, provides the capability to browse, create, modify and delete information in the SDM.

All of these components communicate with one another through Common Object Request Broker Architecture (CORBA) interfaces that adhere to the specifications of the SDM and WFM. The use of CORBA in the SAVE architecture provides two primary benefits:

1. Software vendors develop a single interface for data exchange with other tools as opposed to the point to point interfaces that would be required without the use of CORBA.

2. Implementations allow data storage locations to be defined by the deployment sites, not the software development team.

In addition to the CORBA-compliant portions of the SAVE architecture, SAVE contains two additional features. An electronic collaborative design notebook allows users to communicate with one another and share information that is not part of the common data. This notebook may also be used to collect decision-making history. Three-dimensional CAD data is a key element of many simulation models. The SAVE environment provides a direct link from most major CAD systems for extracting pertinent feature data to both the cost analysis tool and the assembly variability simulation tool. Currently, the CAD data is translated for use in the other tools, but direct links are quickly becoming available.

Chapter 2

Concept of Operations

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

CHAPTER 2 – TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Section **Title** **Page**

LIST OF FIGURES

LIST OF TABLES

Table **Title** **Page**

1.0 Overview

This concept of operations for the SAVE Virtual Manufacturing Environment provides an overview of the system benefits, its components, their integrated function and a typical application scenario.

2.0 Benefits of SAVE

The SAVE integrated tool suite provides a seamless environment for design and manufacturing simulation using a common database for shared data. Within SAVE there is a great opportunity for efficiency improvements through the repeated use of simulation data and results that supports the "build once - use many times" philosophy.

Using SAVE and the commercially available tools within the suite provides the following benefits to the concept design and development process.

- Cost estimation techniques will provide accurate prediction of the fabrication and assembly cost of mechanical parts.

- Simulation of the manufacturing process will allow the identification and resolution of bottlenecks before they affect the schedule, thus reducing span times and overhead costs.

- Simulation and validation tools will make it possible to identify and take corrective action on manufacturing problems very early in the design process, thus providing a significant reduction in design changes.

- Manufacturing simulation will identify potential problems in capabilities and capacity within the planned manufacturing schema.

- Management tools will provide for an integrated process of developing cost, schedule, and risk studies from a common database, thus allowing "what if" studies for the IPPT and management visibility.

2.1 SAVE Impact on JSF

SAVE's use of simulation permits the IPPT to understand the impact of preliminary decisions in a timely manner in order to provide proactive feedback before Engineering Manufacturing Development (EMD) begins. Studies have been conducted within the F-22 program to isolate the anticipated SAVE process improvement over existing procedures. The expected percentage gains in the JSF development and manufacturing processes were extrapolated from the F-22 studies and are shown in Table 2-1.

Table 2-1: Expected EMD Process Improvements for JSF

SAVE METRICS		
METRIC	**SAVINGS**	**OBJECTIVE OF METRIC**
DESIGN CHANGES	28%	Effectiveness of Design Process Control Unwanted Changes
DESIGN TO COST	12%	Meet Production Cost Goals
PROCESS CAPABILITY	5%	Reduce Product Variation Evaluate Key Capabilities
REDUCED FAB & ASSY INSPECTION	6%	Indication of Higher Quality
LEAD TIME REDUCTION	10%	Meet Production Cost Goals
SCRAP, REWORK, REPAIR	11%	Reduce Cost of Non-Quality Evaluate Corrective Action Effectiveness
INVENTORY TURNS	2%	Reduce Cycle Time Reduce Non-value Added Activities

These savings represent a significant impact to the program. When applied during the life cycle of the JSF program, the result is potential cost avoidance of $1B, or about 1-2% of the aircraft's projected life cycle cost.

2.2 SAVE's Impact on Typical EMD Problems

At a lower level, SAVE can impact historical problem areas for EMD programs. Improvements in these areas are due to both the use of the simulation tools themselves and the integrated environment through which they share data.

2.2.1 Late Engineering Release

A significant contributor to the late engineering release problem is the lack of concurrent understanding and development by engineering and manufacturing organizations. This lack of coordination often results in corrective action by engineering before the design can be released. The SAVE system facilitates early product/process decisions to provide clear guidance to the EMD organizations which permits resources to be correctly coordinated and focused in the selected direction. Should the decision parameters change, SAVE can rapidly react through sensitivity studies to develop the optimum response and issue coordinated guidance to the cognizant IPPT members.

2.2.2 Long Lead Times

The consequences of outmoded databases and non-integrated planning process are long lead times. The integrated SAVE system allows the exploration of alternative resource allocation and requirements. The system can address design considerations, make or buy decisions, and the fabrication, assembly, and factory spans to develop alternative optimum processes that will provide the shortest possible lead times. In the event customer or vendor requirements change, materials change, or timing issues develop, SAVE has the ability to assist the IPPT in developing solutions to the lead time problem.

2.2.3 Part Shortages

Using the SAVE tool suite, resource requirements can be rapidly assessed and optimized to provide shortest most cost effective paths. Simulations may be conducted rapidly to determine the principal causes to the shortage and how best to engage a flexible manufacturing system to make process improvements in the factory. Through SAVE, alternative schemes can be analyzed to find which path permits the fastest recovery. The SAVE system can be applied to the inventory management process that simulates the flow of both internal and external part sources, their schedule, on dock status, and critical leverage points.

2.2.4 Factory Personnel Requirements

The ability to use ergonomic, assembly cell, and factory flow simulation tools provides management a realistic means to explore the personnel requirements and then develop the most efficient personnel utilization plan. The SAVE system can conduct sensitivity analyses for different factory operating or environmental scenarios to forecast personnel needs.

2.2.5 Planning Completeness and Clarity

The IPPT early use of the SAVE tool suite expedites correct and concise product planning through the modeling and simulation data capture and visualization process. The planners can complete the manufacturing process plan with a high degree of confidence since they can track and modify the process through simulation. The plans have been developed and verified in an integrated modeling environment prior to beginning the physical manufacturing of EMD components. In addition, the simulations created as part of this process may be deployed to shop personnel as animated work instructions.

2.2.6 Tooling Performance

SAVE's ability to verify design implications, their related processes, and/or planning implementation prior to making parts and tools will reduce problems in the factory. Assembly tool performance can be predicted and validated as part of assembly-cell simulation process.

2.2.7 Introducing New Processes and Procedures

New processes and procedures can be established and validated by SAVE's simulation capability prior to their being introduced into the factory. The implementation of the new methods can be accelerated through the use of repeated simulations as part of training process for the factory employee.

2.2.8 Processes Are Insufficiently Characterized

A major impediment to proficient manufacturing is insufficient process description and limited characterization for the process or procedure user. SAVE's simulation and modeling capabilities support the development, characterization, and verification of new assembly procedures and processes.

2.2.9 Cycle Time Realization to Goal

SAVE enables bottom up simulations that can result in process optimization and improvement at each level in the factory. Factory sensitivity studies can be conducted to isolate key areas that restrict goal realization. From these studies, resource requirements can be rapidly assessed and optimized to provide the shortest most cost effective plan to achieve the goals.

3.0 Components of SAVE Design and Simulation Tool Suite

The integrated SAVE tools are industry leading, off the shelf commercial tools that are normally used as stand-alone environments.

These tools provide cost analysis, dynamic process visualization, planning, reduction of process variability, factory floor layout, production flow analysis, facilities planning, and risk assessment. The tool suite and its integrated environment are shown in Figure 2-1.

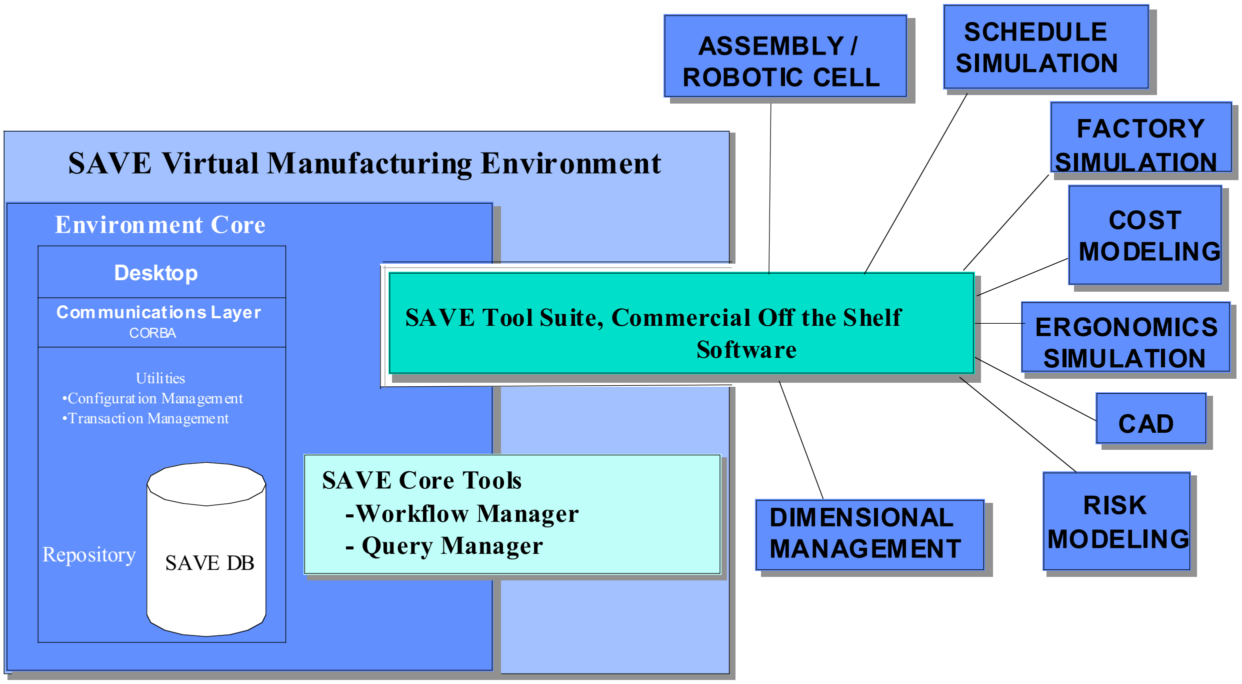


Figure 2-1: SAVE Virtual Manufacturing Environment

These tools are installed to support the needs of different projects and project domains. These tools are integrated with the associated component and model libraries maintained in the SAVE Data Model. The SAVE Data Model permits information to be exchanged between tools in a manner that accommodates tool substitution without impacting information flow from one process step to another.

The SAVE tool suite encompasses the spectrum of virtual manufacturing permitting the engineering and manufacturing team to use one or more of the tools to develop and analyze associated simulation models. The SAVE architectural concept allows users to use their desktop machine (either workstation or PC) to gain access to the available integrated tools and information.

Table 2-2 lists the design, factory, and management tools that were selected for integration into the SAVE environment. This table supplies the type of tool, the SAVE-integrated tool name and developer, and an introductory description of the tool. Even though specific vendor tools are mentioned here for the purpose of this discussion, the SAVE infrastructure supports any tool that fits into these classes. The sections that follow provide a more detailed description of the functions, inter-relationships, and expected usage of these tools in the SAVE environment. Within these sections, the tools are divided into three distinct categories: Design, Factory, and Management.

Table 2-2: The SAVE Tool Suite

Computer Aided Design (CAD) *DASSAULT CATIA:* A 3D design tool widely accepted by major aerospace companies throughout the world. Provides part, assembly, tool, inspection equipment and support equipment designs and NC programs.
Dimensional Management- Part *Engineering Animation VSA-3D:* Statistical simulation analysis to predict the amount of variation that can occur in an assembly due to specified design tolerances.
Assembly Cell Simulation *DENEB IGRIP:* An assembly cell process simulation tool with advanced 3D graphics for visualization. Can be used to provide ergonomics simulation and off-line programming and/or human model interactions.

Factory Process Simulation

DENEB QUEST: Factory process simulation tool for assessing productivity, cost-effectiveness, and efficiency of proposed manufacturing systems. Quest provides a full system for analyzing cycle times, throughput, and factory flow. This system would consider feasible manufacturing alternatives.

Ergonomic Analysis

DENEB ERGO: Simulates and analyzes ergonomic and human factors engineering issues. It provides a visual analysis of a person in a virtual workspace and allows evaluation of access, safety, etc.

Schedule Simulation

SYMIX FACTOR / AIM: Provides capacity design and continuous capacity scheduling through the use of a graphical user interface, database, and simulation technologies. This tool is also used to perform “high level” factory process simulation prior to full CAD model development and help define feasible manufacturing alternatives.

Cost Estimating

COGNITION COST ADVANTAGE: An expert system shell for building cost algorithms which evaluate a design based on features, material and processes. It assigns costs to these attributes and provides a total cost estimate of a part or assembly.

Risk Analysis-Component Level

SAIC ASURE: The Analytical System for Uncertainty and Risk Evaluation (ASURE) is an analytical tool that supports better decision making in any development or acquisition process.

Each tool employed by SAVE represents an independently functioning simulation. The SAVE-enabled linking of the models allows the interaction of key variables and data from one model to another. This ability leverages the strengths of the individual models by permitting complex design, assembly, and factory scenarios to be addressed with a high degree of confidence in the results.

The IPT using the SAVE tools within the SAVE architecture will be able to process information and pass the output to the SAVE Data Model where it will be available as input data for the other SAVE tools. An example is the use of IGRIP/ERGO to simulate an assembly operation to generate span time information. This span time information is used in the QUEST factory simulation for bottleneck analysis, the FACTOR/AIM schedule tool for resource utilization studies, and the cost model for assembly cost estimation. These types of inter-relationships are typical of the entire tool suite. Table 2-3 shows the usage of the SAVE shared data element categories by tool. Figure 2-2 illustrates the flow of this data through the system via the SAVE interface.

Table 2-3: SAVE Tool Suite Shared Data Elements

Data	ASURE	CA	FA	IGRIP	QUEST	VSA	Project
Times/ Durations		1. Calc	3. Use	2. Update	3. Use		3. Use
Schedules/ Dates			2. Update				1. Input 3. Use
Cost		1. Calc					
Quantity Produced	1. Use	1. Use	1. Use		1. Use		
Op Name/ID/Desc	2. Use	2. Use	2. Use	2. Use	2. Use	2. Use	1. Input
Op Precedence	2. Use	2. Use	2. Use 3. Update	2. Use 3. Update	2. Use 3. Update	2. Use 3. Update	1. Input
Op Repetition/Qty	1. Use	1. Use		1. Use	1. Use	1. Use	
Op Type		1. Use				1. Use	
Features	1. Use	1. Use				1. Use	
Part Name/#	2. Use	2. Use	2. Use	2. Use	2. Use	2. Use	1. Input
Part Quantity	1. Use	1. Use	1. Use		1. Use	1. Use	
Part Type	1. Use	1. Use	1. Use	1. Use	1. Use	1. Use	
BOM/ Indenture		1. Use		1. Use	1. Use		1. Input
CAD (Part/Rsrce)				1. Use	1. Use	1. Use	
Materials	1. Use	1. Use				1. Use	
Standard Hours			1. Use	1. Use		1. Use	
Risk (cp,cpk,etc.)	1. Calc 2. Use					1. Calc	
Person Resource (Name,Qty, Skill)		2. Use 4. Use	2. Use 3. Update	2. Use 4. Use	2. Use 3. Update		1. Input 4. Use
Tool Resource (Name,Qty, Type)	2. Use 4. Use	2. Use 4. Use	2. Use 3. Update	2. Use 3. Update	2. Use 3. Update	2. Use 4. Use	1. Input 4. Use
Calendar/Shift			1. Use		1. Use		
Tool Reliability	1. Use					1. Use	

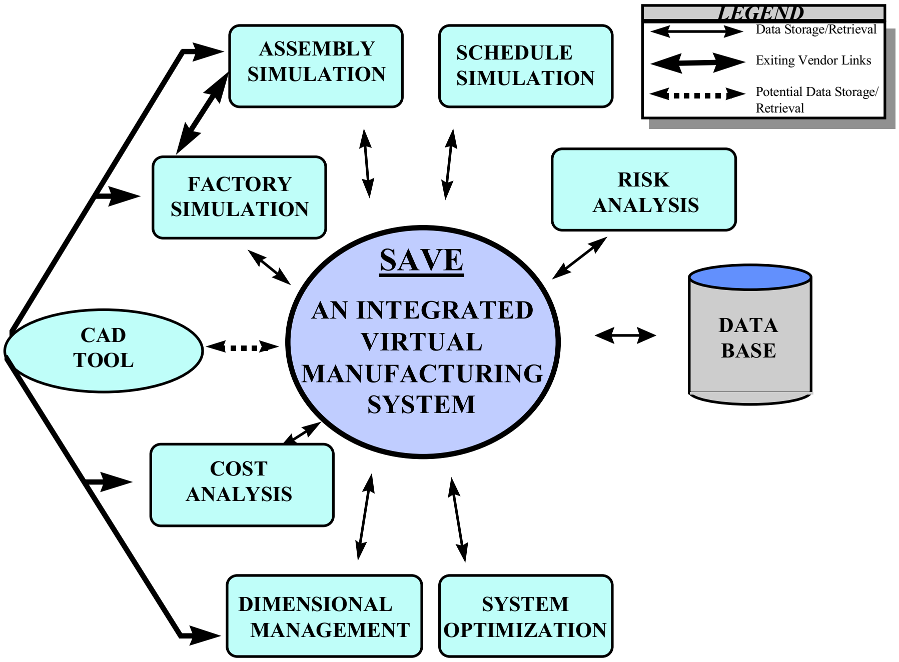


Figure 2-2: SAVE Tool Suite Data Flow

In order to provide a better understanding of the system and its interactions, each tool within the design, factory and management categories in SAVE is described in more detail in the following sections. The descriptions will contain graphics depicting the usage of the tool in a typical development program as well as illustrations showing the tool descriptions with data interactions from sources (inputs) through sinks (outputs). Figure 2-3 shows the format for this tool description.

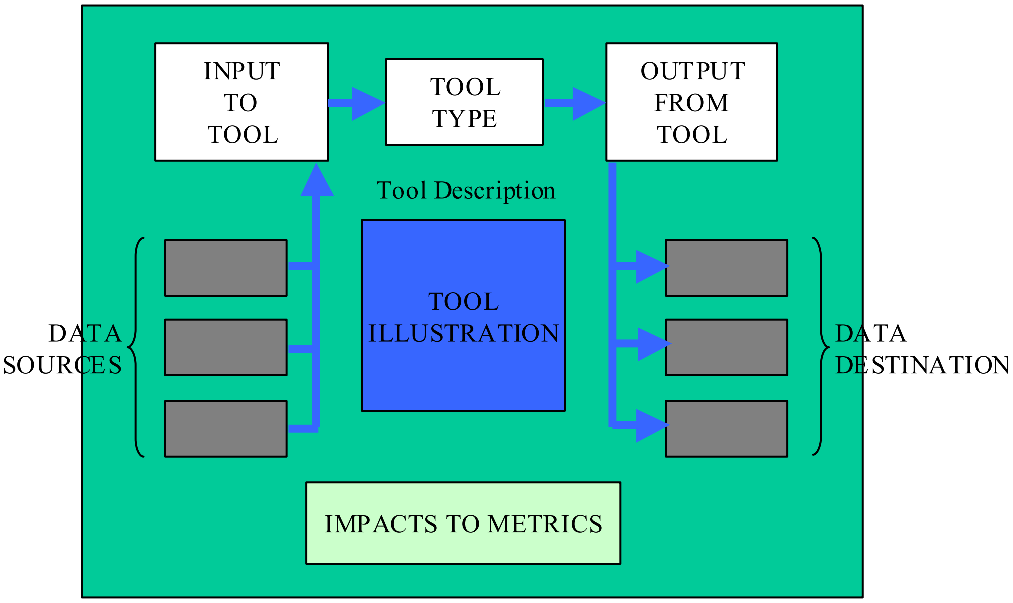


Figure 2-3: Illustration/Description Format for SAVE Tool Suite

3.1 SAVE Design Related Tools

Computer Aided Design and Dimensional Management tools provide 3D design and tolerancing capabilities that are just beginning to be fully exploited in new product development programs.

Computer Aided Design (CAD)

A three-dimensional design tool. Provides part, assembly, tool, inspection equipment and support equipment designs and N/C programs.

Dimensional Management- Part

Statistical simulation analysis to predict the amount of variation that can occur in an assembly due to specific design tolerances.

Figure 2-4 depicts the expected usage of the above tools over the life of a typical Engineering and Manufacturing Development (EMD) program. The segments of the graph are divided along major E&MD milestones as follows: Contract Award to System Design Review (SDR). SDR to Preliminary Design Review (PDR), PDR to Critical Design Review (CDR), CDR to Physical Configuration Audit (PCA), and PCA to Full Production.

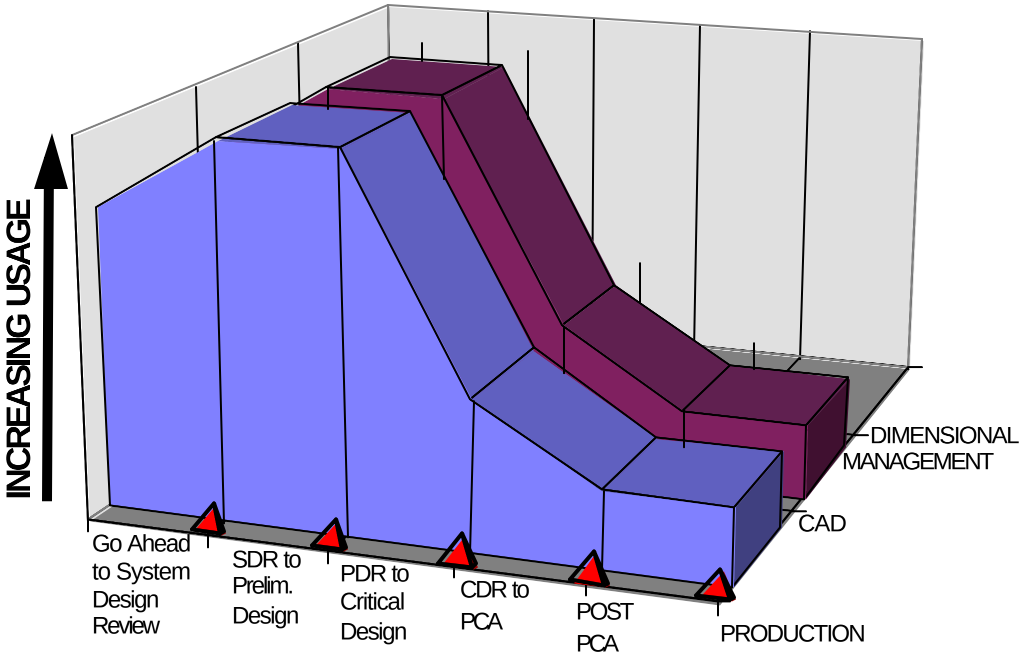


Figure 2-4: Design Tool Usage

3.1.1 Computer Aided Design (CAD) Tool

Almost every major aerospace company in the world uses 3-D design tools. They provide geometric information for part, assembly, tool, special inspection equipment, special support equipment designs and N/C programs.

The Computer Aided Design (CAD) tool is used to generate three-dimensional solid models from part / assembly definitions input by engineers or suppliers. This CAD system provides the geometry for part and assembly designs, tool designs, and N/C programming all from one common source. Additionally, the data generated by CAD will also be used as input to the factory simulation, assembly cell simulation, ergonomics evaluation, assembly and part tolerancing, and cost modeling packages.

CAD's planned utilization during the course of an EMD program is forecasted in Figure 2-5. CAD is normally the first EMD tool used in a development program. The interaction of CAD and the other SAVE tools will have a significant impact on the potential improvement of key metrics.

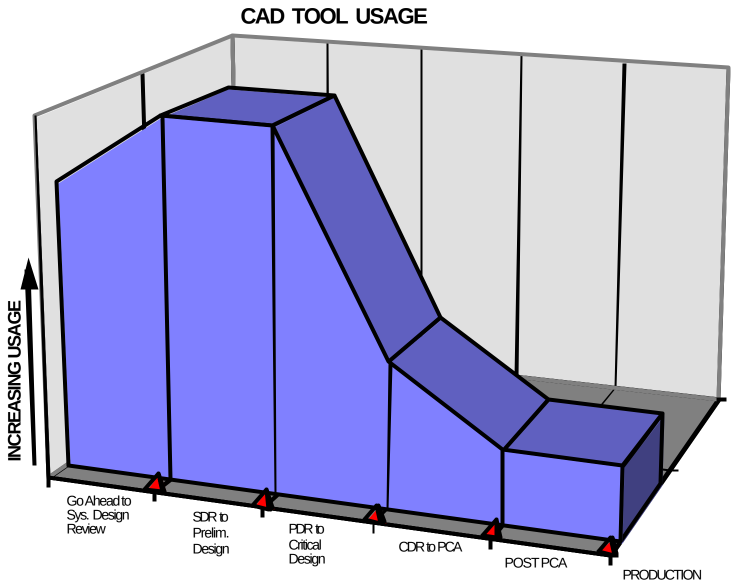
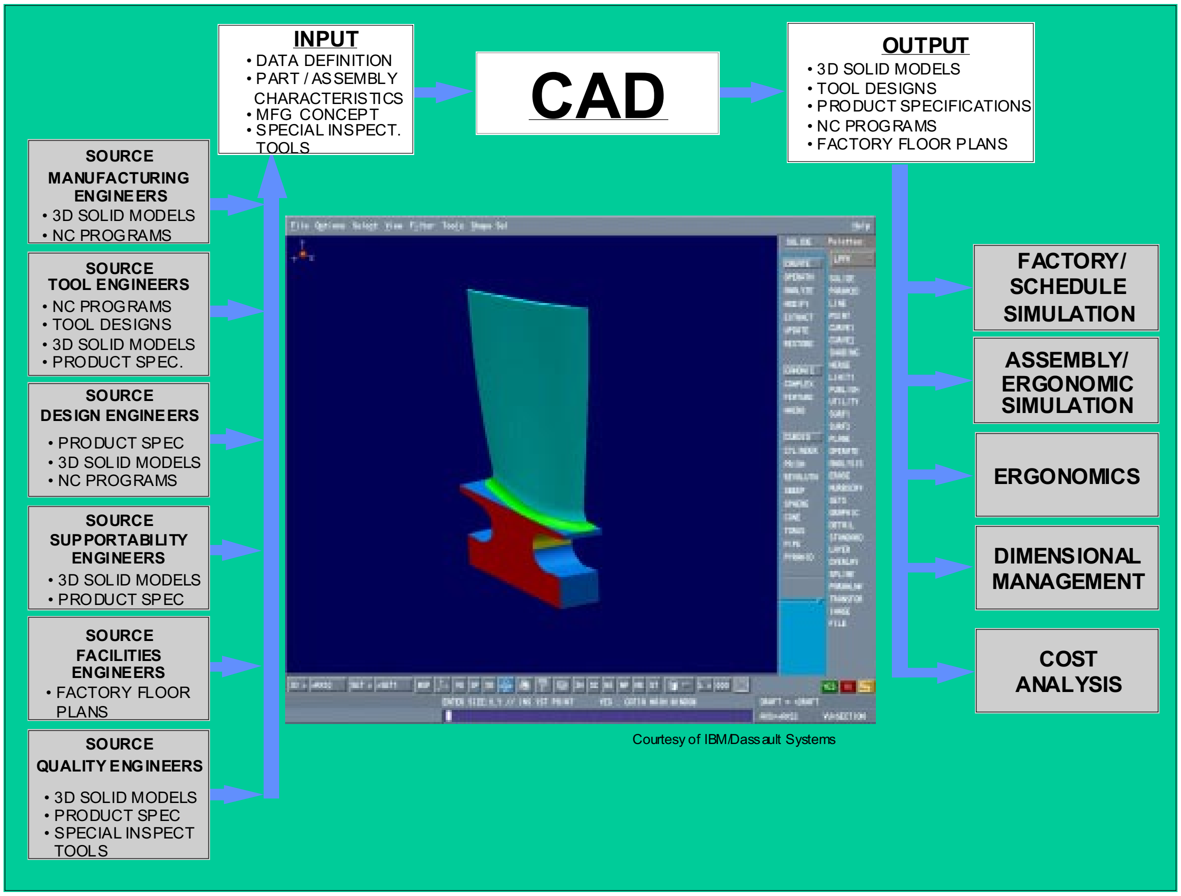


Figure 2-5: CAD Utilization

The sources for the CAD input of part and tool definition, characteristics, and assembly data are shown in Figure 2-6. The primary outputs of CAD include 3D solid models, tool designs, specifications, NC programs, factory floor plans, inspection equipment, and other related product/process information may be inputs to other components of the SAVE tool suite.



Courtesy of IBM/Dassault Systems

Figure 2-6: CAD Tool Interactions

3.1.2 Dimensional Management Tool

This tool mathematically fits a “gage” to the three dimensional model of the part. This “gage” is then used to determine if the design, tooling, manufacturing and assembly processes as specified optimally meet requirements. Part tolerances, datum schemes, assembly process variations and part deflections are used to determine over / under constraints assigned to part tolerances.

The calculated optimum tolerances are available for input for further evaluation by assembly tolerance models, cost models, and schedule and risk simulations. Dimensional management tool utilization during the course of an EMD program is forecasted in Figure 2-7 shown below.

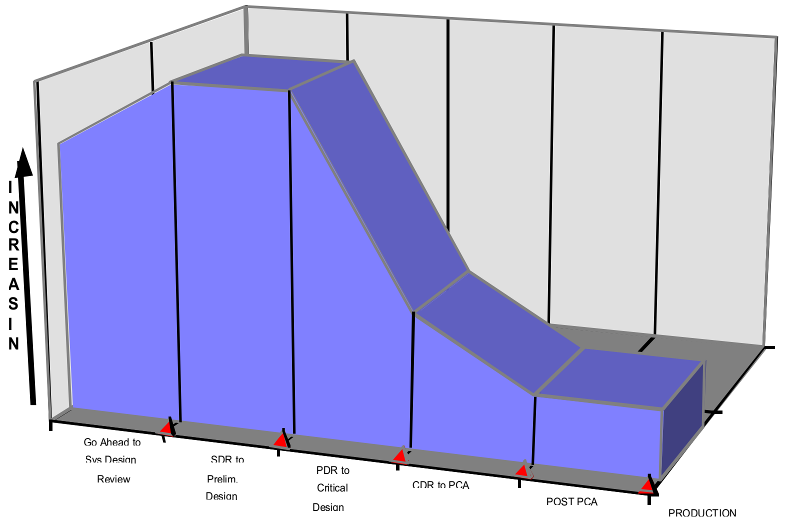
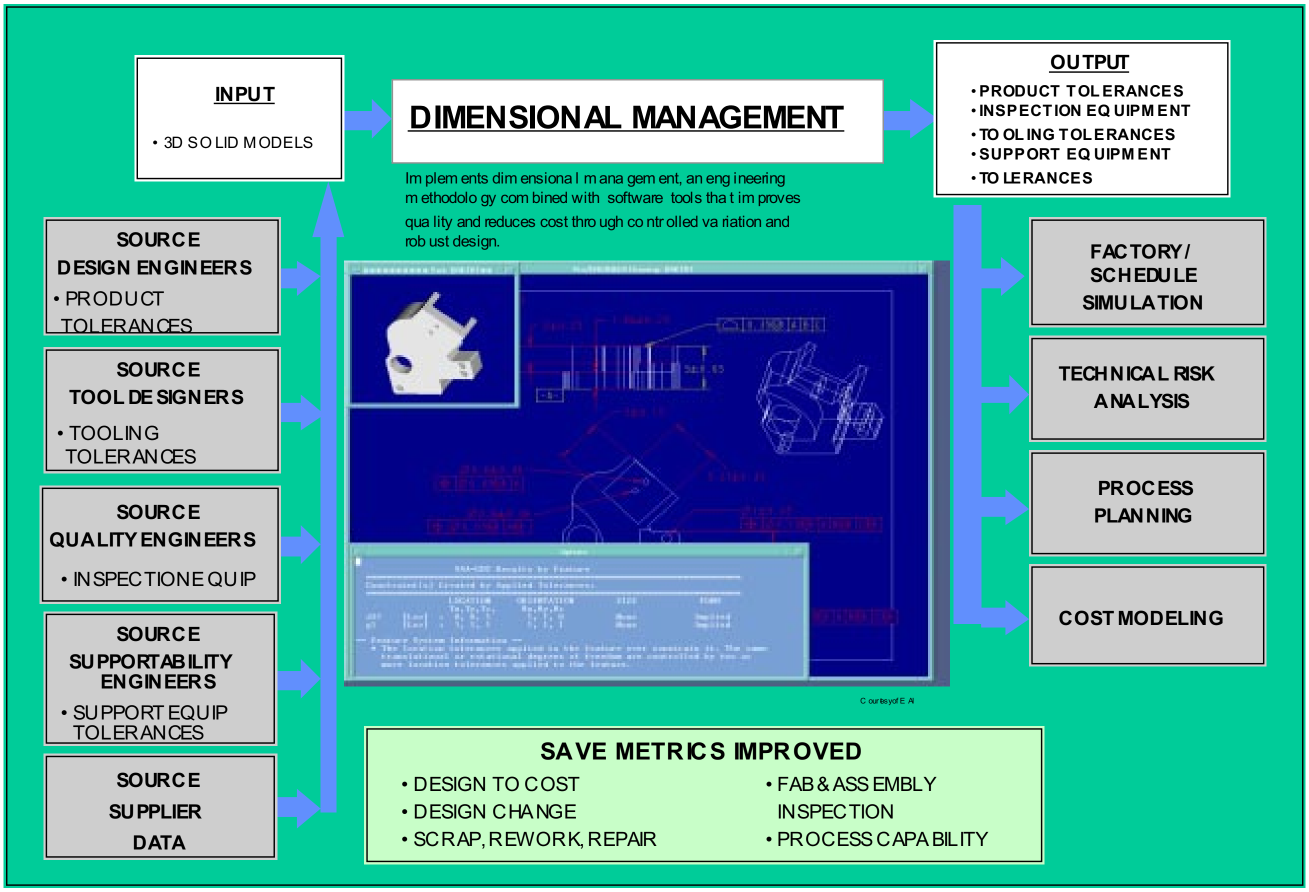


Figure 2-7: Dimensional Management Tool Usage

The sources for the tolerance analysis data are shown in Figure 2-8. The tolerance assessments are conducted as part of the design evolution of components, assemblies, and manufacturing tooling. This software program uses statistical simulation techniques to predict the amount of variation that can occur in an assembly due to specified design tolerances, fixturing tolerances and manufacture/assembly variation.



INPUT

- 3D SOLID MODELS

DIMENSIONAL MANAGEMENT

Implements dimensional management, an engineering methodology combined with software tools that improves quality and reduces cost through controlled variation and robust design.

OUTPUT

- PRODUCT TOLERANCES
- INSPECTION EQUIPMENT
- TOOLING TOLERANCES
- SUPPORT EQUIPMENT
- TOLERANCES

SOURCE DESIGN ENGINEERS
- PRODUCT TOLERANCES

SOURCE TOOL DESIGNERS
- TOOLING TOLERANCES

SOURCE QUALITY ENGINEERS
- INSPECTIONEQUIP

SOURCE SUPPORTABILITY ENGINEERS
- SUPPORT EQUIP TOLERANCES

SOURCE SUPPLIER DATA

FACTORY/ SCHEDULE SIMULATION

TECHNICAL RISK ANALYSIS

PROCESS PLANNING

COST MODELING

Courtsy of EAI

SAVE METRICS IMPROVED

- DESIGN TO COST
- DESIGN CHANGE
- SCRAP, REWORK, REPAIR
- FAB & ASSEMBLY INSPECTION
- PROCESS CAPABILITY

Figure 2-8: Dimensional Management Tool Interactions

3.2 SAVE Factory Related Tools

Assembly, process and ergonomic simulations provide the heart of SAVE's capabilities with respect to manufacturing simulations of shop floor processes.

Assembly Cell Simulation

An assembly cell process simulation tool with advanced 3D graphics for visualization. Can be used to provide ergonomics simulation and off-line programming and/or human model interactions.

Factory Process Simulation

Factory process simulation tool for assessing productivity, cost-effectiveness, efficiency of proposed manufacturing systems, cycle times, throughput, and factory flow analysis.

Ergonomic Analysis

Simulates and analyzes ergonomic and human factors engineering issues. It provides a visual analysis of a person in a virtual workspace and allows evaluation of access, safety, etc.

Figure 2-9 depicts the expected usage of the above tools over the life of the EMD program. The segments of the graph are divided as follows: Contract Award to System Design Review (SDR), SDR to Preliminary Design Review (PDR), PDR to Critical Design Review (CDR), CDR to Physical Configuration Audit (PCA), and Post PCA to Full Production.

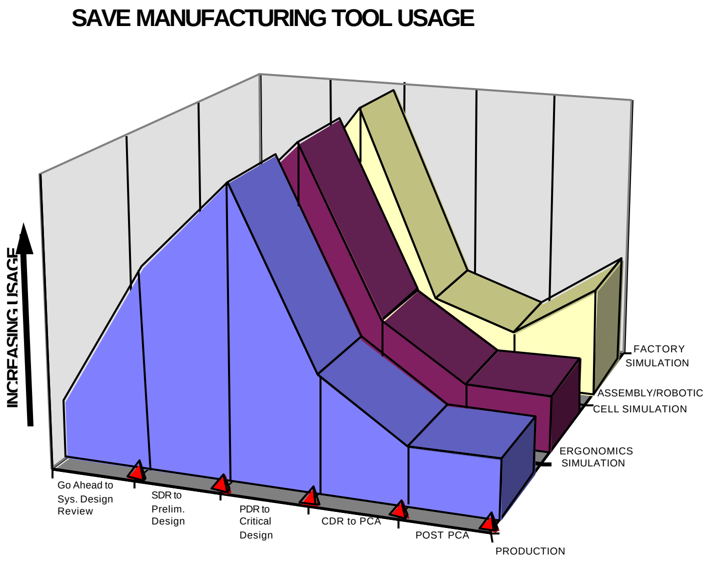


Figure 2-9: Factory Tool Usage

3.2.1 Assembly Cell Simulation Tool

The assembly simulation tool is used to simulate machinery, robots, and human interactions within an assembly work cell. The tool uses data from ergonomics models, and scheduling to verify and prototype concept designs involving structures, mechanical systems and humans. Actual device geometry, motion attributes, kinematics, clamps, fixtures and input / output logic are incorporated to produce simulations.

The resultant data is available for input to scheduling, cost, risk, factory simulations, and production models as a refinement in the entire model feedback loop. Additionally, the completed models allow for simulation based training to be conducted with end-users, who can achieve proficiency in operating and maintaining weapon systems without the associated risk to production schedules. Figure 2-10 depicts the anticipated usage of the assembly cell simulation tool during an EMD program.

Figure 2-11 shows the interactions for the assembly cell simulation tool. This tool uses advanced three-dimensional graphics for visualization and provides an interactive environment in which to verify production concepts, work cell designs, and manufacturing processes before implementing them on the shop floor.

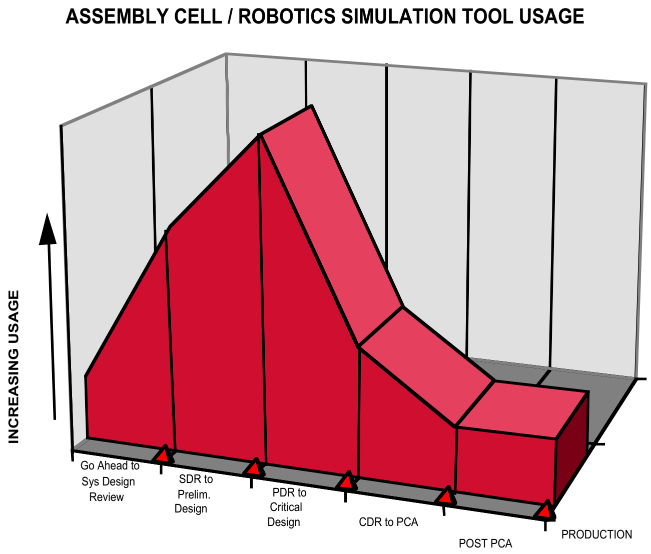
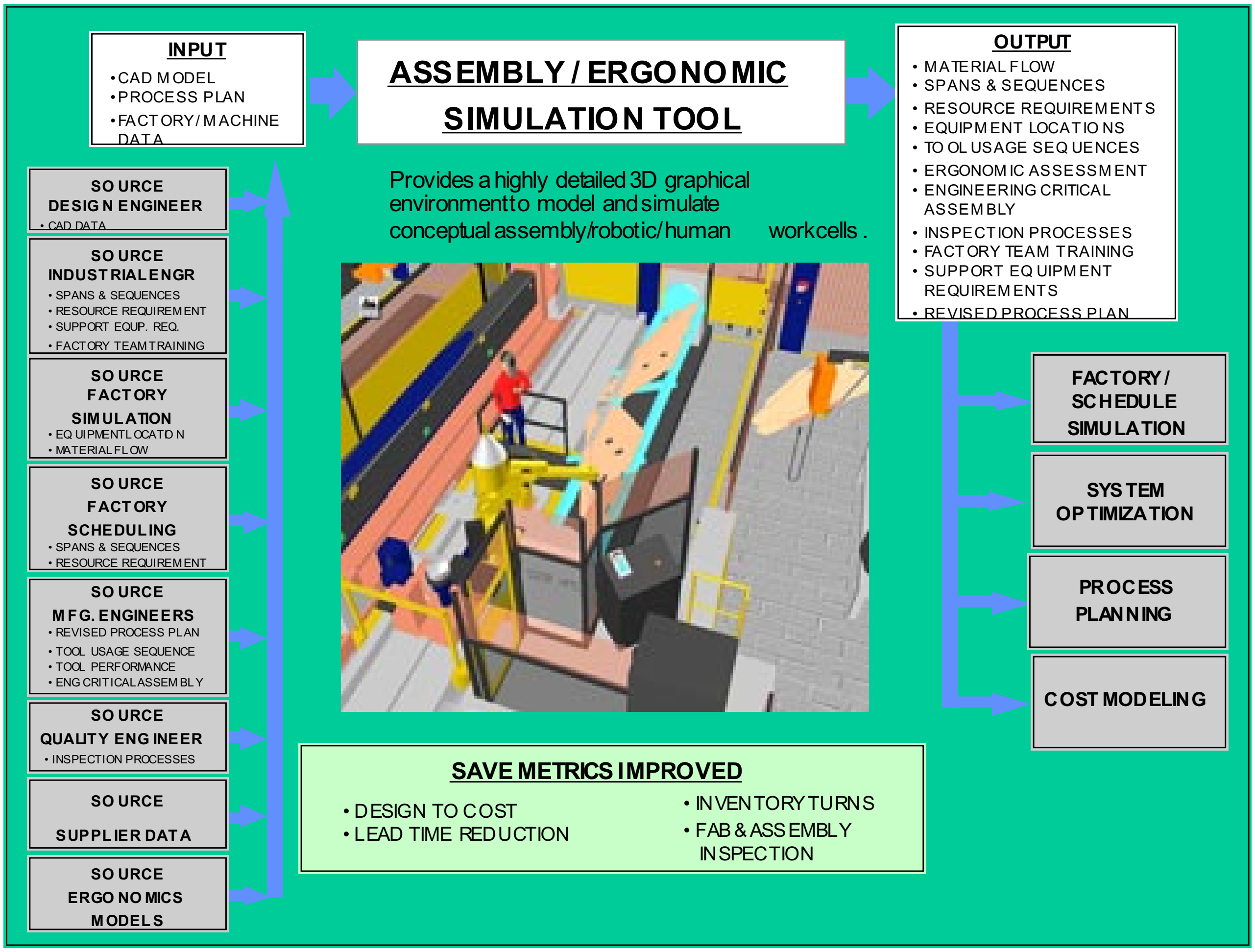


Figure 2-10: Assembly Cell Tool Usage



INPUT

- CAD MODEL
- PROCESS PLAN
- FACTORY/MACHINE DATA

ASSEMBLY / ERGONOMIC SIMULATION TOOL

Provides a highly detailed 3D graphical environment to model and simulate conceptual assembly/robotic/human workcells.

OUTPUT

- MATERIAL FLOW
- SPANS & SEQUENCES
- RESOURCE REQUIREMENTS
- EQUIPMENT LOCATIONS
- TOOL USAGE SEQUENCES
- ERGONOMIC ASSESSMENT
- ENGINEERING CRITICAL ASSEMBLY
- INSPECTION PROCESSES
- FACTORY TEAM TRAINING
- SUPPORT EQUIPMENT REQUIREMENTS
- REVISED PROCESS PLAN

SOURCE DESIGN ENGINEER
- CAD DATA

SOURCE INDUSTRIAL ENGR
- SPANS & SEQUENCES
- RESOURCE REQUIREMENT
- SUPPORT EQUP. REQ.
- FACTORY TEAMTRAINING

SOURCE FACTORY SIMULATION
- EQUIPMENTLOCATDN
- MATERIALFLOW

SOURCE FACTORY SCHEDULING
- SPANS & SEQUENCES
- RESOURCE REQUIREMENT

SOURCE MFG. ENGINEERS
- REVISED PROCESS PLAN
- TOOL USAGE SEQUENCE
- TOOL PERFORMANCE
- ENG CRITICAL ASSEMBLY

SOURCE QUALITY ENGINEER
- INSPECTION PROCESSES

SOURCE SUPPLIER DATA

SOURCE ERGONOMICS MODELS

FACTORY / SCHEDULE SIMULATION

SYSTEM OPTIMIZATION

PROCESS PLANNING

COST MODELING

SAVE METRICS IMPROVED

- DESIGN TO COST
- LEAD TIME REDUCTION
- INVENTORY TURNS
- FAB & ASSEMBLY INSPECTION

Figure 2-11: Assembly Cell Tool Interactions

3.2.2 Ergonomics & Human Factors Simulation Tool

The ergonomics simulation model uses Operational Safety and Health Administration (OSHA) and industry guidelines regarding energy expenditure, lifting capacity, and posture analysis integrated with human factors engineering to simulate worker movements within the workplace.

It allows a visual analysis of a person in a virtual work environment designed and scaled to simulate the environment. The IPPT can explore alternative set-ups to evaluate their respective issues such as access, safety, time requirements and work capacity before committing to a particular arrangement.

The findings of this model act as direct inputs into the scheduling model, the assembly-cell, factory simulation, risk, and production models.

The ergonomics simulation model utilization that is forecasted for an EMD program is shown in Figure 2-12.

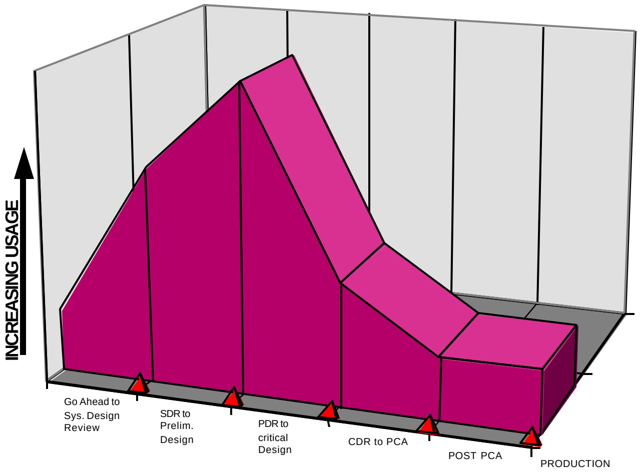
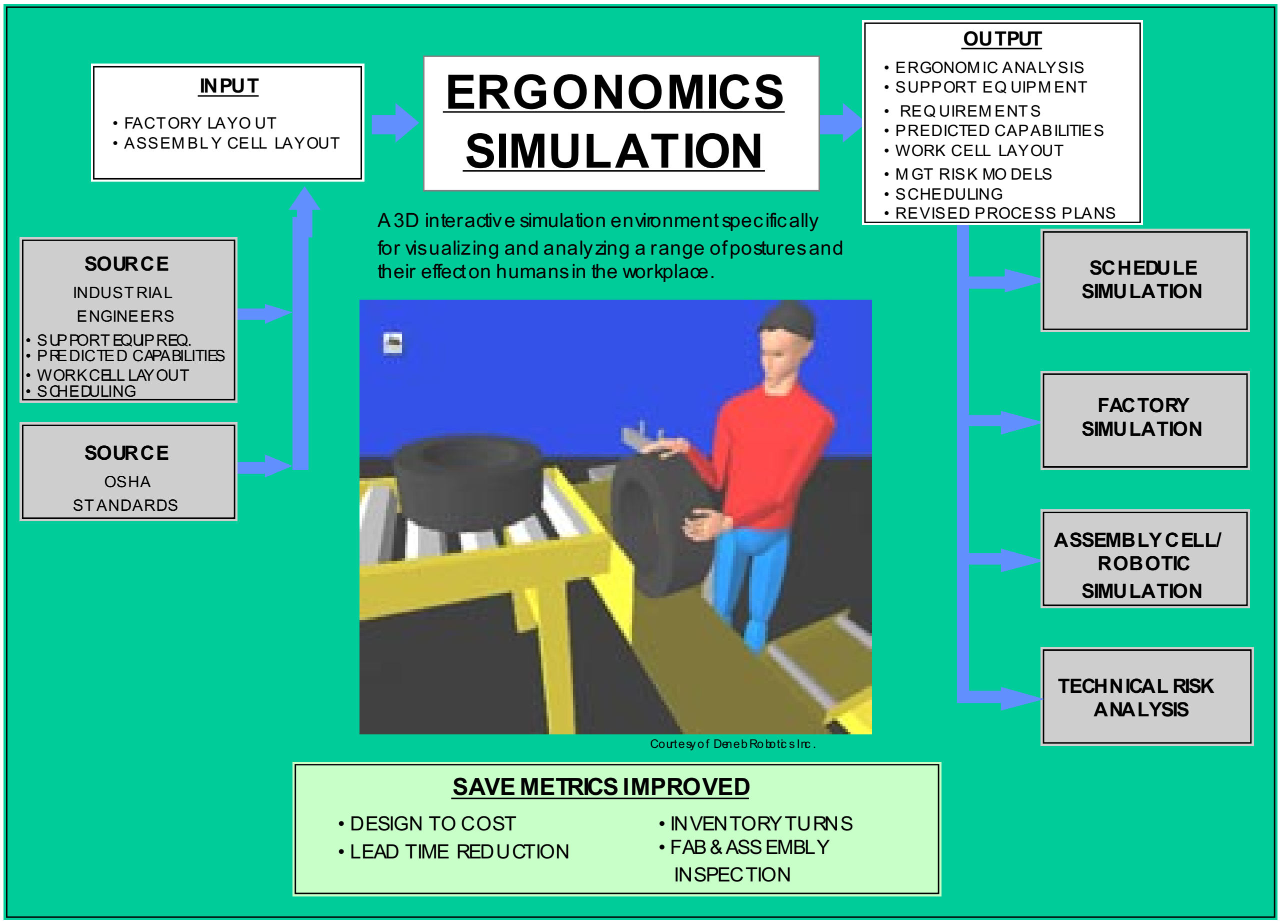


Figure 2-12: Ergonomics Tool Usage

The human factors specialist can determine if a worker's anthropometry will allow him/her to work comfortably in an existing workplace. The industrial engineer can use ergonomics simulation to check the workplace design to ensure the worker can complete their job in the allocated time. This engineer can also use ergonomics simulation to test time standards for new tasks and improve time standards for existing jobs. Figure 2-13 shows the inputs and interactions for the ergonomics simulation tool.



INPUT

- FACTORY LAYOUT
- ASSEMBLY CELL LAYOUT

ERGONOMICS SIMULATION

OUTPUT

- ERGONOMIC ANALYSIS
- SUPPORT EQUIPMENT
- REQUIREMENTS
- PREDICTED CAPABILITIES
- WORK CELL LAYOUT
- MGT RISK MODELS
- SCHEDULING
- REVISED PROCESS PLANS

SOURCE
INDUSTRIAL
ENGINEERS

- SUPPORT EQUIP REQ.
- PREDICTED CAPABILITIES
- WORKCELL LAYOUT
- SCHEDULING

SOURCE
OSHA
STANDARDS

A 3D interactive simulation environment specifically for visualizing and analyzing a range of postures and their effect on humans in the workplace.

Courtesy of Deneb Robotics Inc.

SCHEDULE SIMULATION

FACTORY SIMULATION

ASSEMBLY CELL/ ROBOTIC SIMULATION

TECHNICAL RISK ANALYSIS

SAVE METRICS IMPROVED

- DESIGN TO COST
- LEAD TIME REDUCTION
- INVENTORY TURNS
- FAB & ASSEMBLY INSPECTION

Figure 2-13: Ergonomics Tool Interactions

3.2.3 Factory Simulation

The factory simulation tool is used for assessing productivity, cost-effectiveness and efficiency of proposed manufacturing systems. It is an interactive, three-dimensional graphical simulation tool that allows users to generate, or import from CAD, geometry to represent the factory system. This enables users to quickly build, modify and optimize simulation models of their individual operations.

Factory simulation may be used in an interactive graphical mode, or it may be run merely generating a listing of the results. It provides a full system for analyzing the difference in production rates between proposed alternatives, and how those alternatives will affect the financial operation of the system.

The factory simulation utilization that is forecasted for an EMD program is shown in Figure 2-14 below.

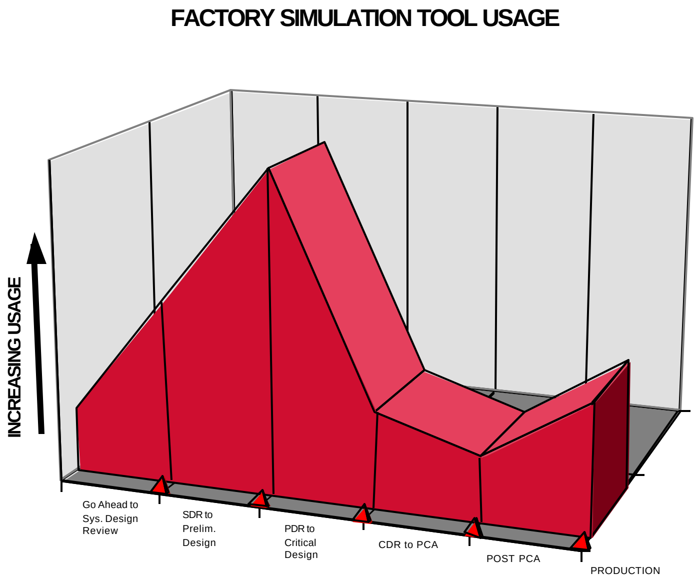
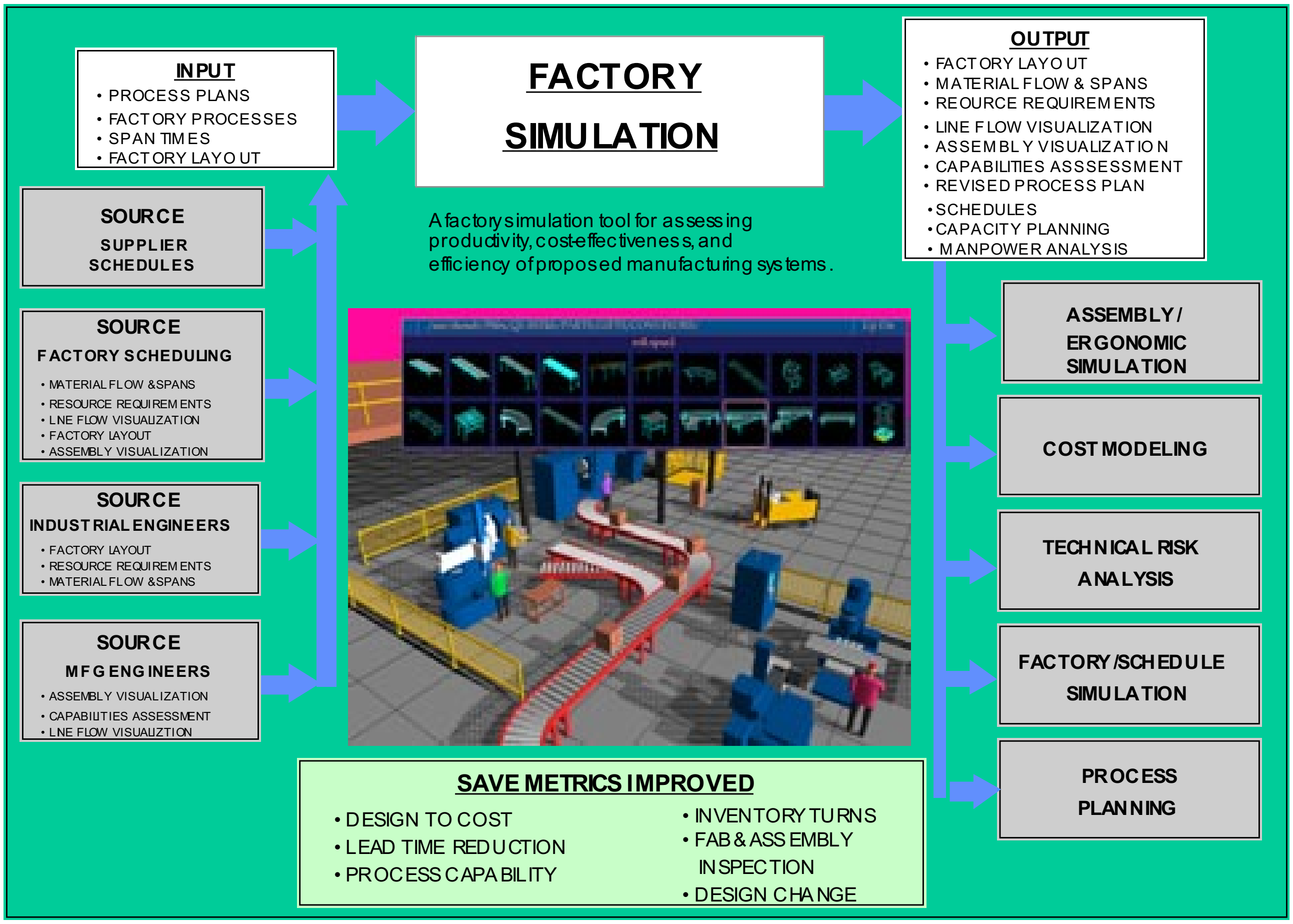


Figure 2-14: Factory Simulation Tool Usage

This tool directly emulates real-world system behaviors that are associated with each resource, including factory flow activities, routing, sequencing, and merging. It also permits the manufacturing representatives on the IPPT to test alternative factory layouts in the course of developing the optimum operation.

The results generated by the factory simulation model may be used as direct feedback into the scheduling systems, the assembly-cell systems, and the risk, production and cost models. The interactions of the factory simulation tool are shown in Figure 2-15.



INPUT
- PROCESS PLANS
- FACTORY PROCESSES
- SPAN TIMES
- FACTORY LAYOUT

FACTORY SIMULATION

OUTPUT
- FACTORY LAYOUT
- MATERIAL FLOW & SPANS
- REOURCE REQUIREMENTS
- LINE FLOW VISUALIZATION
- ASSEMBLY VISUALIZATION
- CAPABILITIES ASSSESSMENT
- REVISED PROCESS PLAN
- SCHEDULES
- CAPACITY PLANNING
- MANPOWER ANALYSIS

A factory simulation tool for assessing productivity, cost-effectiveness, and efficiency of proposed manufacturing systems.

SOURCE
SUPPLIER SCHEDULES

SOURCE
FACTORY SCHEDULING
- MATERIAL FLOW & SPANS
- RESOURCE REQUIREMENTS
- LNE FLOW VISUALIZATION
- FACTORY LAYOUT
- ASSEMBLY VISUALIZATION

SOURCE
INDUSTRIAL ENGINEERS
- FACTORY LAYOUT
- RESOURCE REQUIREMENTS
- MATERIAL FLOW & SPANS

SOURCE
MFG ENGINEERS
- ASSEMBLY VISUALIZATION
- CAPABILITIES ASSESSMENT
- LNE FLOW VISUALIZTION

ASSEMBLY / ERGONOMIC SIMULATION

COST MODELING

TECHNICAL RISK ANALYSIS

FACTORY /SCHEDULE SIMULATION

PROCESS PLANNING

SAVE METRICS IMPROVED
- DESIGN TO COST
- LEAD TIME REDUCTION
- PROCESS CAPABILITY
- INVENTORY TURNS
- FAB & ASSEMBLY INSPECTION
- DESIGN CHANGE

Figure 2-15: Factory Simulation Tool Interactions

3.3 SAVE EMD Management Related Tools

Schedule, risk, and cost estimating tools provide the basis for the assessment capability within the SAVE tool suite. These tools aid management in optimizing their decision-making process.

Schedule Simulation

Provides capacity design and continuous capacity scheduling through the use of a graphical user interface, database, and simulation technologies to build, simulate capacity planning, logistics, production scheduling and schedule management.

Risk Analysis-Component Level

Provides information regarding consequences associated with decisions. This information allows for guidance in decision making in development or acquisition processes.

Cost Estimating

An expert system shell for building cost algorithms which evaluate a design based on features, material and processes. It assigns costs to these attributes and provides a total cost estimate for a part or assembly.

Figure 2-16 depicts the expected usage of the above tools over the life of the EMD program. The segments of the graph are divided as follows: Contract Award to System Design Review (SDR), SDR to Preliminary Design Review (PDR), PDR to Critical Design Review (CDR), CDR to Physical Configuration Audit (PCA).

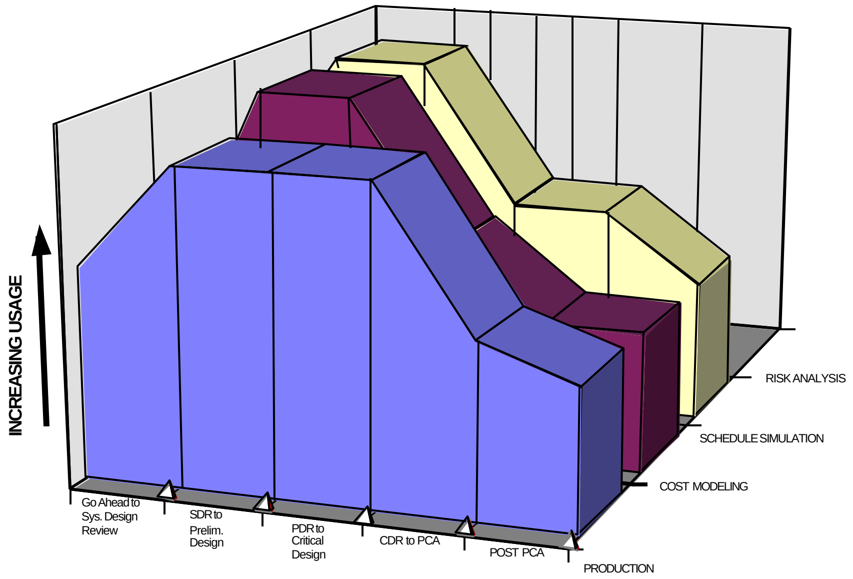


Figure 2-16: Management Related Tool Utilization

3.3.1 Schedule Simulation

Scheduling is critical to all phases of an EMD program. By inputting factors, required delivery dates, and milestones from planning and pricing systems; manpower analysis, resource requirements, availability, capacity design and planning information may be produced.

This is accomplished through the use of a graphical user interface (GUI), database, and simulation technologies that build and simulate logistics and production. All this data provides input for the factory and assembly cell simulations, ergonomic models, cost models, risk models and production models.

The schedule simulation utilization that is forecasted for an EMD program is shown in Figure 2-17 below.

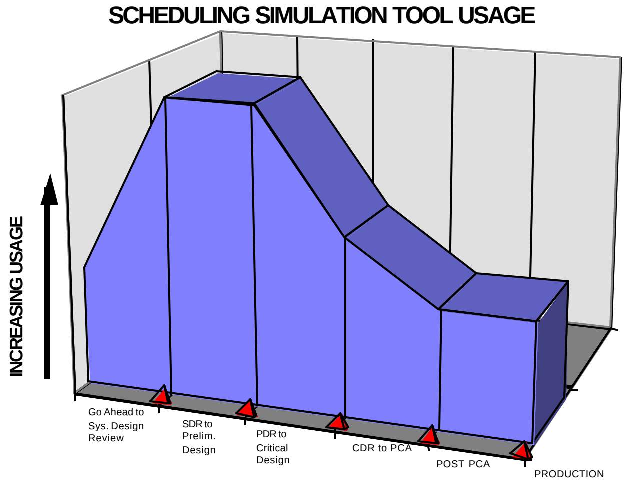
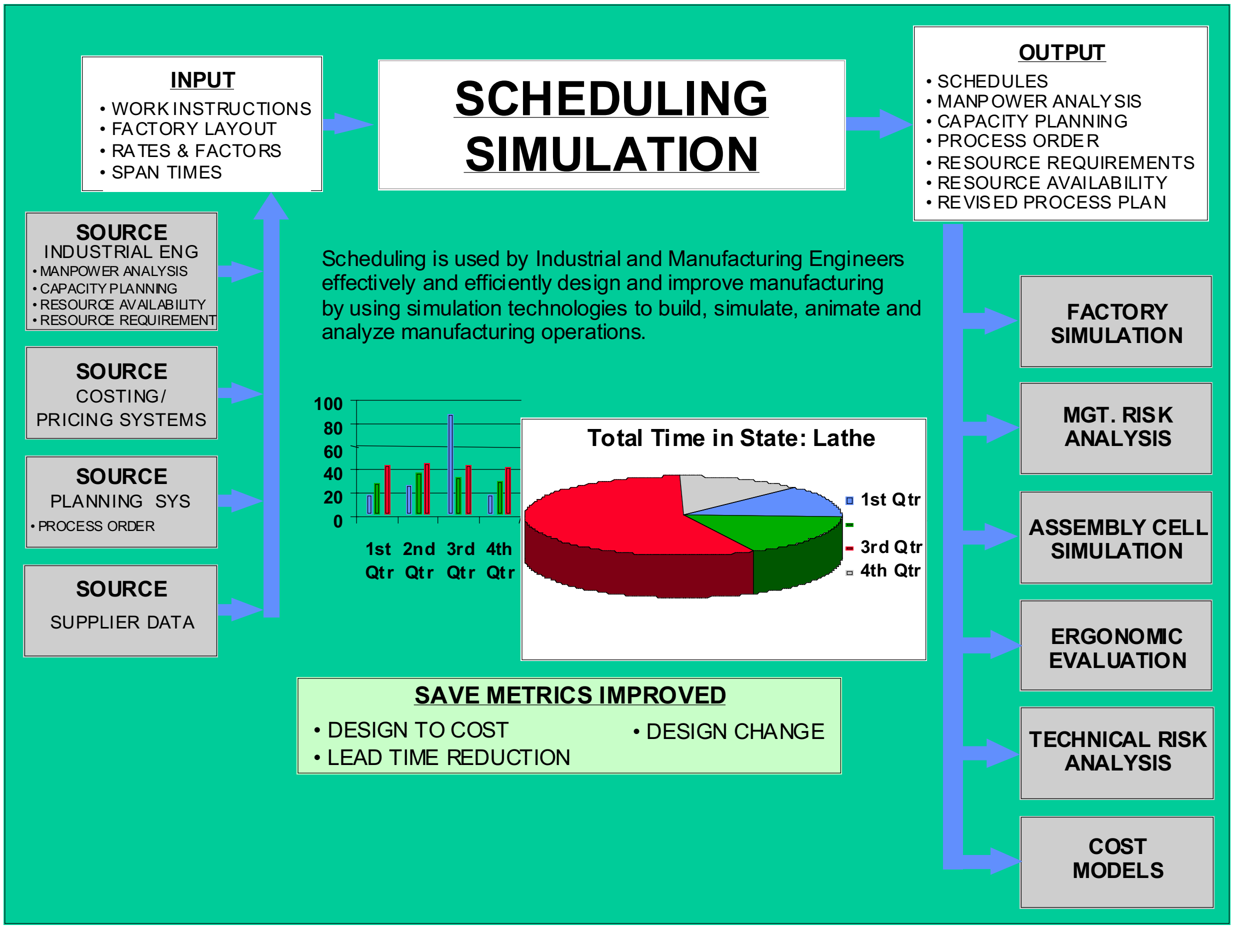


Figure 2-17: Schedule Simulation Tool Usage

The scheduling function is built around a relational database that stores the manufacturing operation description and simulation output. The description of the manufacturing process used by the scheduling package can be created through its own interface or by using existing manufacturing data.

Scheduling models are built graphically and are animated automatically. Animations show machine, operator, fixture, buffer, and part status. Material handling equipment status is shown accurately with displays of transporter system's movement and accumulation. When vehicles are transporting loads, the fixtures and the parts they contain are shown.

Figure 2-18 shows the inputs and interactions of the schedule simulation tool.



INPUT

- WORK INSTRUCTIONS
- FACTORY LAYOUT
- RATES & FACTORS
- SPAN TIMES

SCHEDULING SIMULATION

OUTPUT

- SCHEDULES
- MANPOWER ANALYSIS
- CAPACITY PLANNING
- PROCESS ORDER
- RESOURCE REQUIREMENTS
- RESOURCE AVAILABILITY
- REVISED PROCESS PLAN

SOURCE
INDUSTRIAL ENG

- MANPOWER ANALYSIS
- CAPACITY PLANNING
- RESOURCE AVAILABILITY
- RESOURCE REQUIREMENT

SOURCE
COSTING/
PRICING SYSTEMS

SOURCE
PLANNING SYS

- PROCESS ORDER

SOURCE
SUPPLIER DATA

Scheduling is used by Industrial and Manufacturing Engineers effectively and efficiently design and improve manufacturing by using simulation technologies to build, simulate, animate and analyze manufacturing operations.

SAVE METRICS IMPROVED

- DESIGN TO COST
- DESIGN CHANGE
- LEAD TIME REDUCTION

FACTORY SIMULATION

MGT. RISK ANALYSIS

ASSEMBLY CELL SIMULATION

ERGONOMIC EVALUATION

TECHNICAL RISK ANALYSIS

COST MODELS

Figure 2-18: Schedule Simulation Tool Interactions

3.3.2 Risk Analysis

The risk analysis package is designed to assist the IPPT in incorporating uncertainty factors into their system or concept evaluation decisions when there may be limited data upon which to base the decision. The computational element establishes input distributions, performs uncertainty propagation, and generates confidence profiles for each selected decision criteria. For example, if cost is a key driver in the decision process, the risk analysis will focus on the cost values and the variables that may effect them. The confidence profiles portray the output uncertainty on a variable (decision criterion) as a cumulative distribution on the values. These profiles can include the probability density function, cumulative probability distribution, etc. These distributions may be used as direct input to the production model risk analysis.

The graphic in Figure 2-19 shows the risk simulation tool use for an EMD program is below.

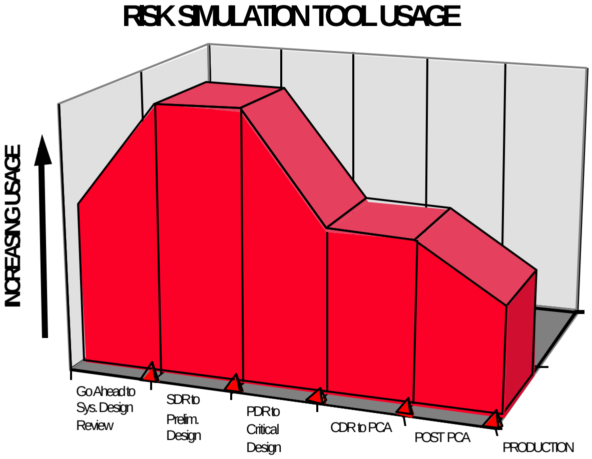
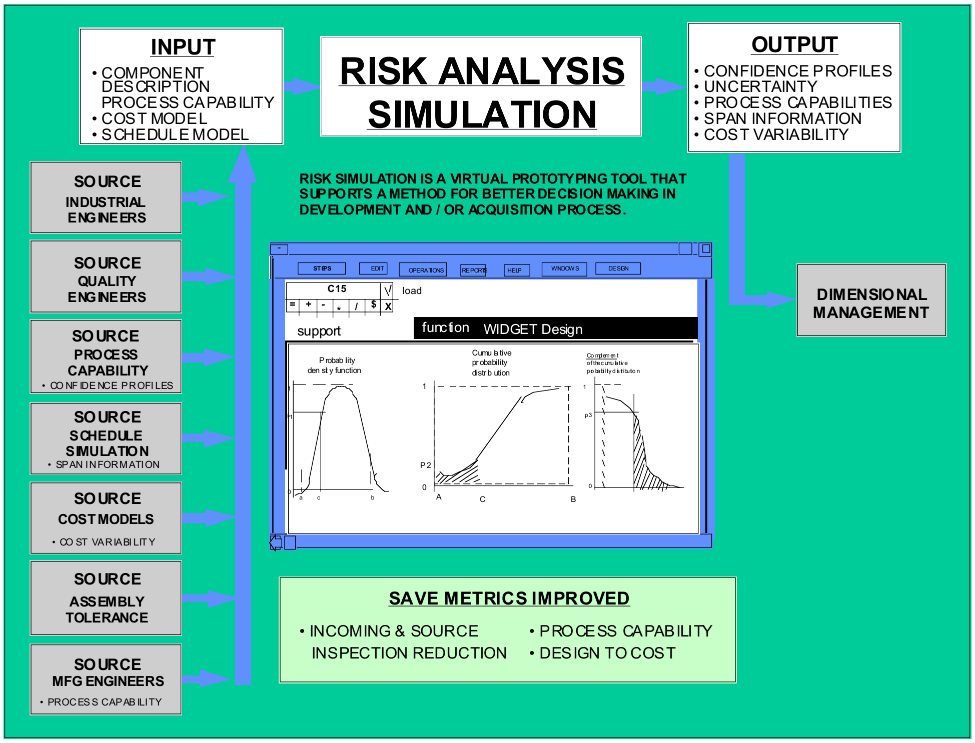


Figure 2-19: Risk Simulation Tool Usage

This risk estimation tool permits the capture and management of evolving system concepts. It allows easy access to the available data for effective analysis and uncertainty management. This tool assists decision makers in incorporating risk and uncertainty in system or concept evaluation decisions based on limited test data and science-based models for estimating system characteristics.

The risk tool permits the capture and management of evolving system concept description. It provides the IPPT and program management easy access to risk related data for analysis and uncertainty management.

Figure 2-20 shows the inputs and interactions for the risk analysis tool.



INPUT
- COMPONENT DESCRIPTION PROCESS CAPABILITY
- COST MODEL
- SCHEDULE MODEL

RISK ANALYSIS SIMULATION

OUTPUT
- CONFIDENCE PROFILES
- UNCERTAINTY
- PROCESS CAPABILITIES
- SPAN INFORMATION
- COST VARIABILITY

SOURCE INDUSTRIAL ENGINEERS

SOURCE QUALITY ENGINEERS

SOURCE PROCESS CAPABILITY
- CONFIDENCE PROFILES

SOURCE SCHEDULE SIMULATION
- SPAN INFORMATION

SOURCE COST MODELS
- COST VARIABILITY

SOURCE ASSEMBLY TOLERANCE

SOURCE MFG ENGINEERS
- PROCESS CAPABILITY

RISK SIMULATION IS A VIRTUAL PROTOTYPING TOOL THAT SUPPORTS A METHOD FOR BETTER DECISION MAKING IN DEVELOPMENT AND / OR ACQUISITION PROCESS.

DIMENSIONAL MANAGEMENT

STEPS
EDIT
OPERATIONS
REPORTS
HELP
WINDOWS
DESIGN
C15
= + - * / $ X
load
support
function WIDGET Design
Probability density function
1
P1
0
a
c
b
Cumulative probability distribution
1
P2
0
A
C
B
Complement of the cumulative probability distribution
1
p3
0

SAVE METRICS IMPROVED
- INCOMING & SOURCE INSPECTION REDUCTION
- PROCESS CAPABILITY
- DESIGN TO COST

Figure 2-20: Risk Analysis Tool Interactions

3.3.3 Cost Simulation Tool

The cost modeling tool is an expert system shell that supports the building of cost and producibility algorithms that evaluate a design based on features, materials, resource requirements from process simulators and process plans for the part / assembly. It assigns a pre-determined cost to each of these attributes and provides a total roll-up cost estimate of the part or assembly. Estimates are available to designers while creating a design, and for team resource evaluations supporting major program producibility and cost decisions.

The cost modeling tool will directly link with the CAD function to make evaluations based directly on design criteria for generating a variety of outputs at the various stages of the decision making process. Additionally, shared process plan and process time data can be shared from process simulators to provide high fidelity cost data that can incorporate production rate as well as span.

The graphic in Figure 2-21 shows the cost simulation tool use for an EMD program is below.

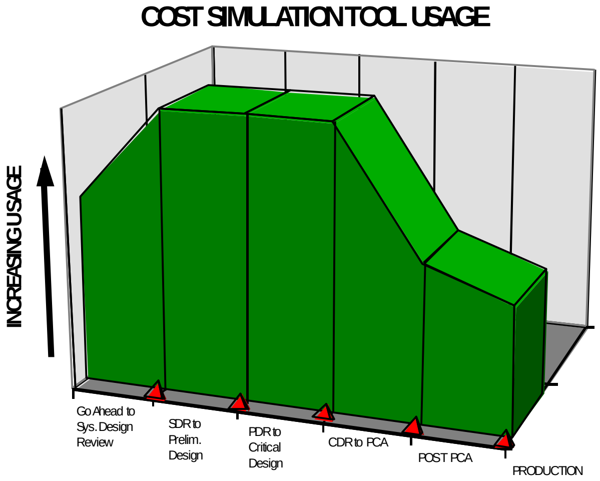
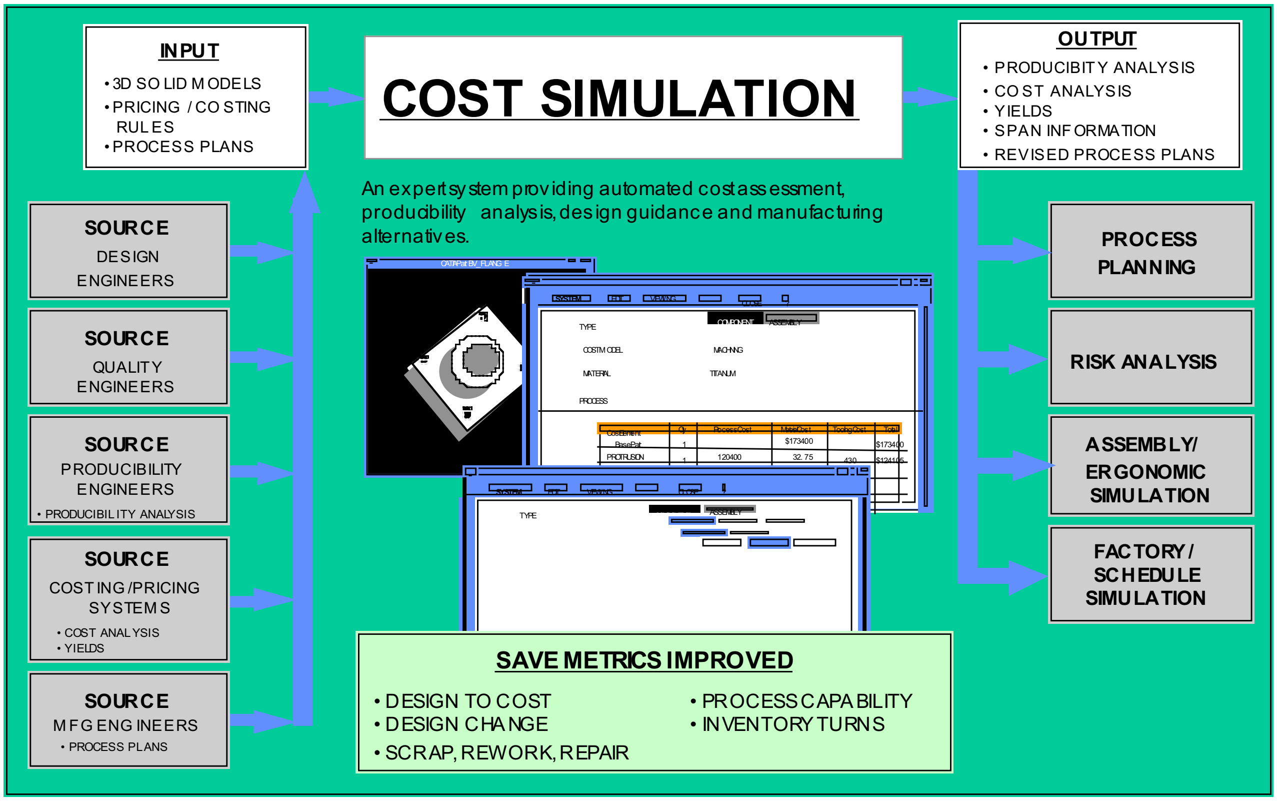


Figure 2-21: Cost Simulation Tool Usage

This tool is a Design for Manufacturing (DFM) expert system that initially provides high level cost data, design guidance, and producibility analysis. It captures manufacturing process knowledge throughout the design cycle and leverages that information to identify costs drivers through all stages of a product's life cycle.

The inputs and outputs of the cost modeling tool are directly integrated with the risk package, assembly-cell and production simulation packages. The cost modeling tool interactions are shown in Figure 2-22.



INPUT

- 3D SOLID MODELS
- PRICING / COSTING RULES
- PROCESS PLANS

COST SIMULATION

OUTPUT

- PRODUCIBITY ANALYSIS
- COST ANALYSIS
- YIELDS
- SPAN INFORMATION
- REVISED PROCESS PLANS

SOURCE
DESIGN ENGINEERS

SOURCE
QUALITY ENGINEERS

SOURCE
PRODUCIBILITY ENGINEERS
- PRODUCIBILITY ANALYSIS

SOURCE
COSTING/PRICING SYSTEMS
- COST ANALYSIS
- YIELDS

SOURCE
MFG ENGINEERS
- PROCESS PLANS

An expert system providing automated cost assessment, producibility analysis, design guidance and manufacturing alternatives.

PROCESS PLANNING

RISK ANALYSIS

ASSEMBLY/ ERGONOMIC SIMULATION

FACTORY / SCHEDULE SIMULATION

SAVE METRICS IMPROVED

- DESIGN TO COST
- DESIGN CHANGE
- SCRAP, REWORK, REPAIR
- PROCESS CAPABILITY
- INVENTORY TURNS

Figure 2-22: Cost Modeling Tool Interactions

4.0 Typical SAVE Scenario

SAVE's virtual manufacturing environment is a comprehensive modeling and simulation environment where the optimum production fabrication / assembly can be determined after being simulated with multiple engineering and manufacturing variables. The full scope of SAVE's virtual manufacturing environment supports all phases of system development including concept design, demonstration / validation, engineering / manufacturing development, and production.

The principal users of the SAVE system are the IPPTs, who will carry out detailed simulation and modeling to evaluate design and manufacturing process alternatives that directly impact risk, cost, and schedule. SAVE's comprehensive tool suite facilitates the detection of problem areas, limitations, and design errors using these simulations and provides guidance to avoid any adverse impact on manufacturing cost, schedule or risk elements. These three elements can actually become competing objectives during the analytical process. When matching engineering, manufacturing, and business requirements, the relative priorities of cost, schedule or risk will change depending on the situation. The ability to share common data between tools enables the IPPT team members to quickly evaluate many scenarios and optimize the priority element (cost, schedule or risk) and minimize the impact on the other elements. The SAVE program provides the ability to develop integrated evolutionary manufacturing plans for complex EMD programs.

As depicted in Figure 2-23, the SAVE concept with its' associated common database allows a virtual collaborative effort by all IPPT team members in any linked remote site.

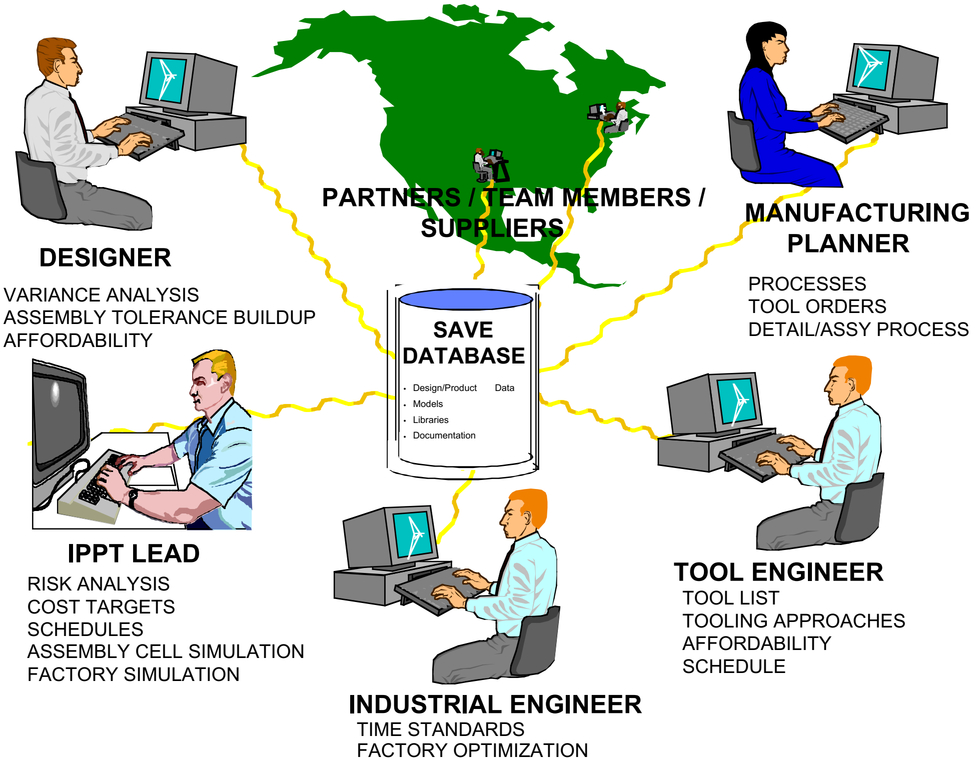


Figure 2-23: SAVE's Virtual Collaborative Environment

This effort can include the entire EMD team and selected suppliers. This close cooperative working relationship throughout the design process is the keystone of the IPPT concurrent engineering effort. Among the benefits of this strategy are thorough assessment of producibility and maintainability capabilities and minimizing or eliminating expensive design and tooling changes.

Key parts of the SAVE system are the architecture, configuration control, and electronic documentation tools that ensure that everyone in the IPPT team is using the proper baseline, constraints, assumptions, and current data sets when performing their part of the overall analysis.

The following scenario discusses a typical SAVE tool use sequence that is representative of a modification program and/or new design effort.

4.1 Requirements Analysis

The new design or change is defined in terms of performance and engineering requirements. The first element of developing an evolutionary EMD plan is the early identification and analysis of potentially responsive concepts through an iterative process of requirements analysis, synthesis, and sizing. Figure 2-24 shows a typical requirements flowdown for this type of development activity.

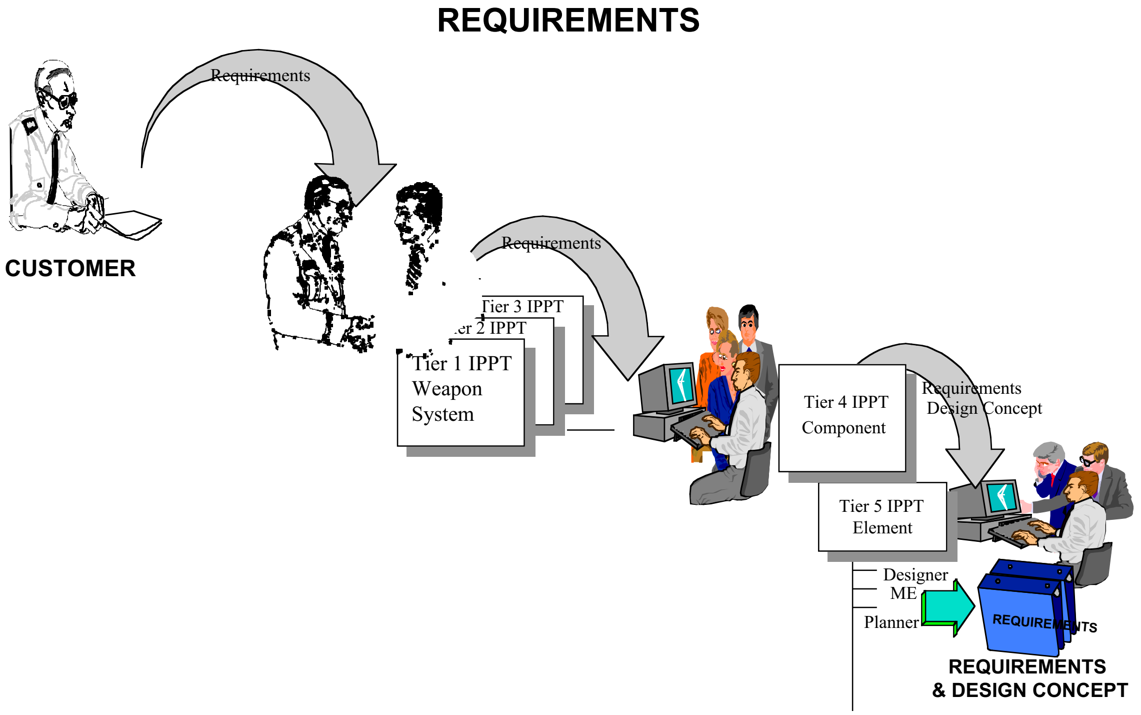


Figure 2-24: Customer Requirements Flow Down To IPPT

The evaluation process for each potential concept begins with the Engineering organizations constructing alternative conceptual design models in the CAD tool. In addition to target performance, the manufacturing objectives of cost, schedule and risk are prioritized into hierarchical objectives and become part of the requirements list for each concept alternative. This information is used by the IPPT to develop design and process alternatives that will potentially satisfy the requirements. Once options are developed the IPPT may employ the virtual manufacturing environment to assess those alternatives relative to the objectives.

Figure 2-25 presents the hierarchical objectives of cost, schedule, and risk that would be used to evaluate each engineering alternative. Although all three objectives are important, the IPPT would determine the relative priority for each of these three objectives. This flowdown format is applied to each engineering alternative, which supports design of simulation experiments and the identification of appropriate response variables.

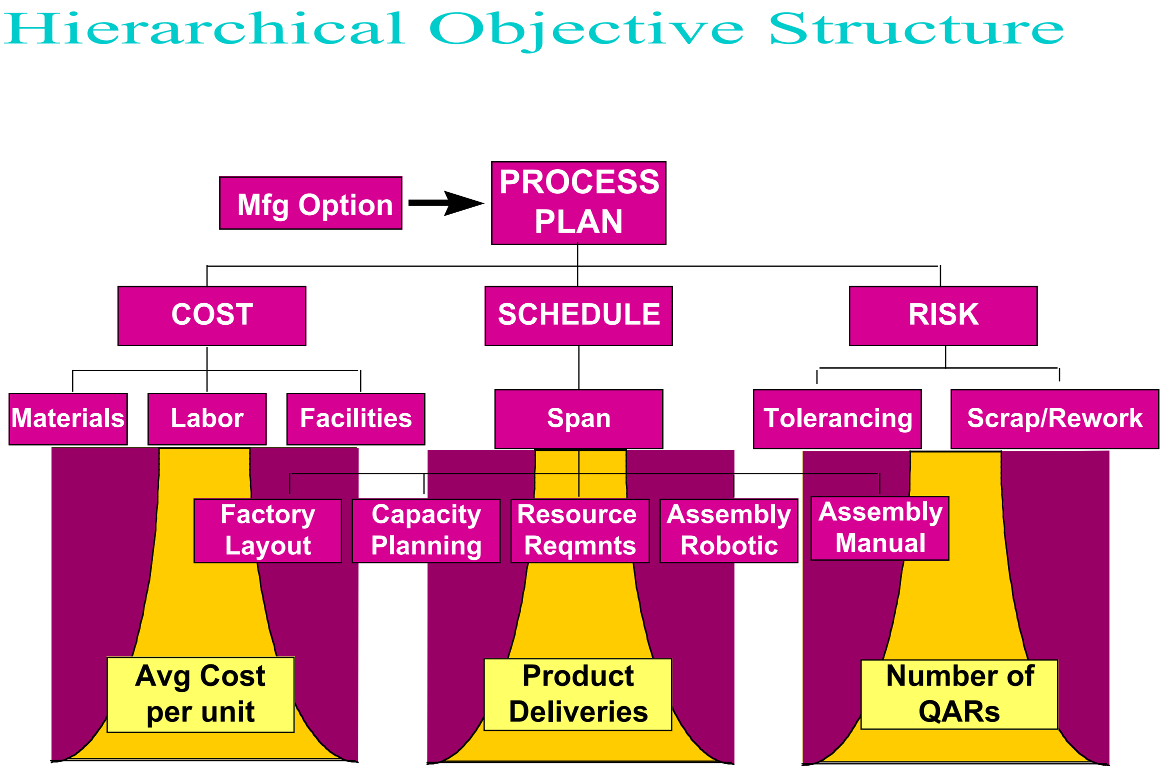


Figure 2-25: Hierarchical Arrangement of Cost, Schedule, and Risk

4.2 Examine Component Data

In the case of a design change, the historical manufacturing data is used to define the baseline for the modification process. This includes any existing CAD models, figures for total costs- including activity cost breakdowns, schedules- broken down by spans for each activity, quality and risk data detailing scrap, rework and repair and maintainability data.

In the case of a new design, data from similar designs, independent studies, and vendor quotes provide baseline data for the new design.

In either case, once the preliminary baselines are established for the concepts, the integrated SAVE EMD process uses the same tool suite and evolving SAVE data sets from contract award through production. The order and precedence of SAVE tool usage and the depth within the supply chain, however, will change as the program progresses.

4.3 Concept Analysis

The feasible design alternatives are refined through a series of initial trade studies and analyses. Simultaneously, preliminary goals and metrics relative to cost, schedule, quality and risk mitigation are established. The ultimate goal of this endeavor is to choose a responsive set of functional alternative(s) that can be designed and manufactured on schedule with minimum cost and risk.

The Manufacturing Planner, Manufacturing/Producibility Engineer, Tooling Engineer, Facilities Engineer work with the Designer to ensure the alternative components and assemblies can be efficiently manufactured. Manufacturing experts use the modified, current, or new design CAD models to develop assembly cell simulation models. Using minimal CAD information, the visualization and simulation processes can identify many of the implications for each concept alternatives.

The engineering requirements, CAD, and manufacturing data serve as inputs to the SAVE tool suite to perform cost simulation, schedule simulation, and risk simulation. The CAD and manufacturing data are used to develop an initial manufacturing process plan with critical path process steps, “minimum crew” manpower requirements, minimum tool, and minimum capital equipment requirements. Using the schedule tool, “high level” simulations are performed to develop preliminary resource requirements (manpower, workstations, major tooling), generate preliminary schedules, and produce the initial manufacturing process plan. This initial schedule output provides input to the cost-estimating tool for preliminary cost evaluation. Using the preliminary schedules and the cost evaluation, an initial risk assessment can be performed.

During this initial phase of the analysis, it is likely that several simulation scenarios using the schedule tool will be required for each engineering option to get a reasonable match between schedule and resource requirements (line balancing). By understanding the key driving variable, use of DOE (design of experiments) methodology can be used to evaluate cost, risk, and schedule in a structured manner. Once cost, risk, and schedule have been evaluated using the SAVE modeling and simulation tools, sufficient information to the IPPT will be available to select the final acceptable configuration, or at least eliminate the majority of the contenders. This data flow expansion is depicted in Figure 2-26.

The best solution depends on the relative priorities of performance, risk, cost, and schedule. SAVE provides the capability to rapidly run a series of different analyses with different cost, risk, and schedule premises for each alternative. The IPPT will take advantage of the tool suite’s capability to simulate, evaluate, and select the individual alternative that is most responsive to the program drivers. One of the advantages of the integrated tool set is that concept model development and virtual manufacturing models can be developed concurrently during the conceptual design process. This is a tremendous advantage to the analytical fidelity of all design alternatives since the expensive and time consuming process of model development and engineering analysis using "point solutions" typically would not be performed on "apparent" high risk or cost inefficient designs.

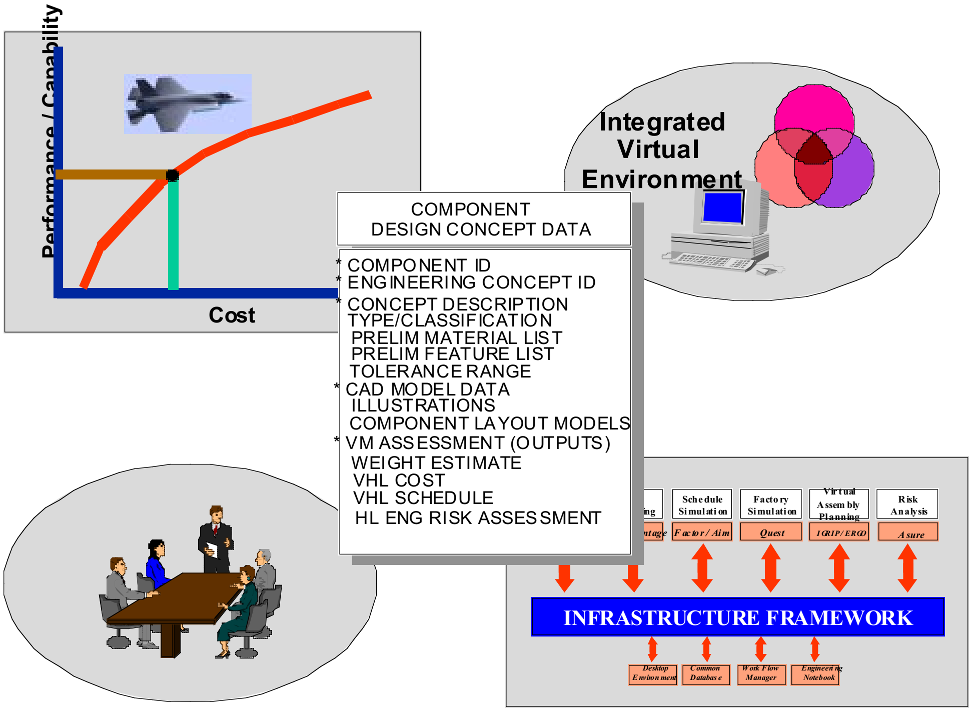


Figure 2-26: Component Concept Data and SAVE Infrastructure

The IPPT makes the final selection after conducting virtual manufacturing analysis for each of the proposed design alternatives. Before selection, each design is evaluated in terms of engineering performance, materials, tolerance boundaries, manufacturing ramifications, cost implications, risk mitigation, and schedule. At this point, the optimum manufacturing approach can be further refined through capacity analysis, automation and ergonomic analysis, and facility and procurement optimization. Additionally, the use of animation tools can be used for management presentations to promote "visualization" during the review and decision making phase.

4.4 Concept Selection

SAVE provides the tools to develop a traceable analysis that permits the IPPT to rationally select the optimum design concept. By using a structured method of analyzing cost, risk and schedule with the SAVE tool set and data base, the engineering concept(s) best meeting the engineering requirements as well as being affordable can be efficiently identified and validated.

The design, manufacturing process plans, and tooling designs for the concept(s) are ready to enter into the preliminary design segment of the program evolution. Along with the preliminary design effort, the IPPTs continue the modeling and simulation process required to complete an approved baseline manufacturing process plan for the concept(s). This plan includes initial process selection, tooling requirements, make or buy decisions, assembly sequencing, installation sequencing along with preliminary manufacturing cost, schedule and risk assessment data. The

IPPTs and program management will have clear insight into the manufacturing plan and the associated cost, risk, and schedule data.

Once the feasible engineering alternatives have been identified, the Preliminary Manufacturing Plans are developed for each feasible alternative and the SAVE virtual manufacturing tool set is used to systematically evaluate each alternative (Figure 2-27).

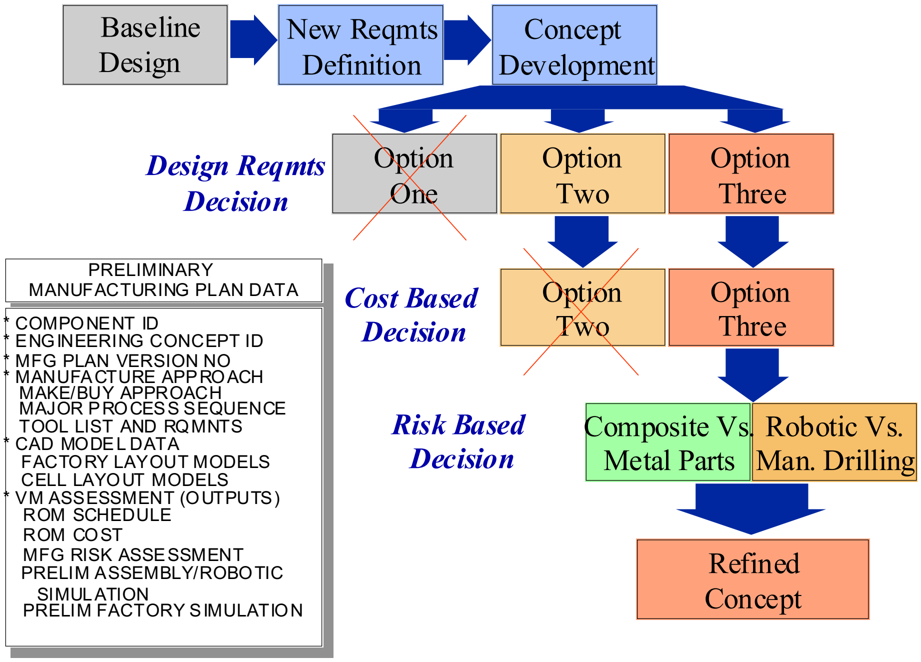


Figure 2-27: Options Are Filtered by SAVE Tool Suite

4.5 Using The SAVE Tool Suite

All of the SAVE tools work in concert to visualize, modify, update, evaluate, and document the design's evolution. The results of this evolving analysis are documented in the collaborative engineering notebook, which allows the IPPT to move forward with a collective understanding of the overall EMD project. In order to discuss each tool's contribution to the program, the scenario is stopped in time where the current manufacturing plan is about to be modified or released based on the outcome of that particular tool's analysis. The engineering activities related to preliminary design are not within the scope of this document, but obviously, they are an integral part of the IPPT process.

4.5.1 Cost Analysis

A cost assessment will be conducted using the preliminary design, a proposed manufacturing process plan, and existing process resource data (manheads, # tools, # workstations, Capital equipment) for each manufacturing alternative. Figure 2-28 presents the primary data elements for performing assembly cost analysis.

The cost tool provides initial information to the simulation/scheduling tool, which includes minimum crew size, required tool set, critical path, and peripheral labor requirements. The process simulation tool returns information including process steps, time for each process step, overall span time, manpower requirements, rate tool requirements, and simulation statistics. The detail component, assembly models, and geometric attribute data are provided by the CAD system. The cost analysis result will be an update to the baseline cost data detailing overall assembly-level costs and itemized cost analysis for the fabrication of detail components. Cost analysis is an iterative process that is modified each time a SAVE tool changes a cost related process variable. The key "cost drivers" will be identified by the cost analysis and opportunities for improvement and cost reduction will be documented in the electronic notebook.

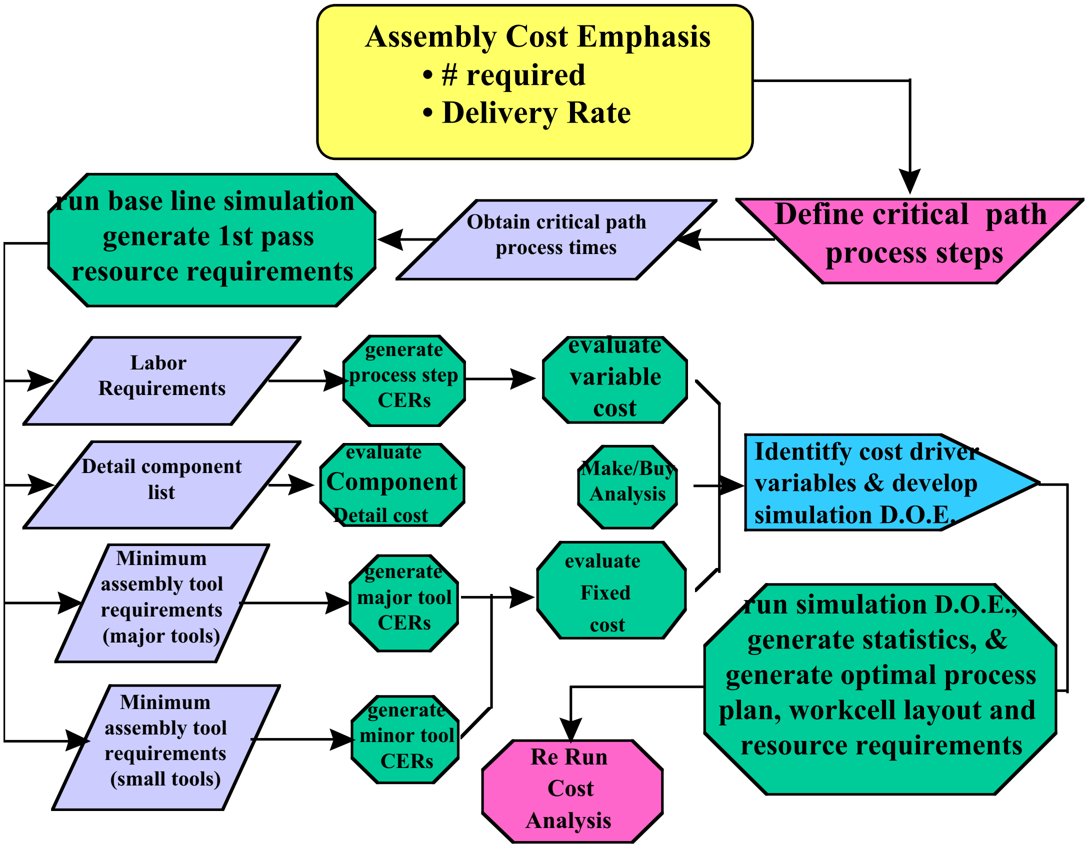

Assembly Cost Emphasis
• # required
• Delivery Rate
Define critical path process steps
Obtain critical path process times
run base line simulation generate 1st pass resource requirements
Labor Requirements
generate process step CERs
evaluate variable cost
Detail component list
evaluate Component Detail cost
Make/Buy Analysis
Identitfy cost driver variables & develop simulation D.O.E.
Minimum assembly tool requirements (major tools)
generate major tool CERs
evaluate Fixed cost
run simulation D.O.E., generate statistics, & generate optimal process plan, workcell layout and resource requirements
Minimum assembly tool requirements (small tools)
generate minor tool CERs
Re Run Cost Analysis

Figure 2-28: Data Elements for Performing Cost Analysis

4.5.2 Assembly Cell Assessment

Using the CAD model, the baseline manufacturing process plan, rate, and machining data from the legacy knowledge base, a preliminary assembly cell model is constructed. The ergonomics and assembly cell simulation models are the principal assembly cell evaluation tools.

These optimized assembly cell layouts are then simulated to provide more accurate span time data, which are included in the next version of the EMD's manufacturing process.

4.5.3 Factory Assessment

A factory simulation model is constructed by applying the ergonomics and assembly cell simulation model(s), factory data, and the updated manufacturing process plan. This model provides a visualization of the factory assembly area (footprint) layout including the machinery type(s), size, number required, and location of machinery.

The tool simulates the effects of sequencing / timing on the efficiency of the system, potential line bottlenecks, layout improvements, inventory storage, and suppliers' delivery schedules. The simulation can assess the impacts of machinery breakdowns and servicing schedules. The tool assists the IPPT and program management in determining the requirements for the optimal output levels to meet schedules.

For facilities with existing functional production lines, the factory simulation model can be used to provide a detailed evaluation of the impact of the new plan's requirements on existing operations. When evaluating options where new facilities are planned, the factory simulation model will enable Facilities Engineers to layout the factory in the most efficient manner. In either case, the factory simulation model provides higher fidelity input to the cost, risk, and scheduling models.

4.5.4 Schedule Assessment

An updated scheduling simulation will be generated from the enhanced manufacturing process plan and the more detailed span time data developed by the ergonomics, assembly cell, and factory simulation outputs. The results of this iteration of the schedule simulation will provide data to the IPPT for identifying areas of potential improvements in the manufacturing process plan, assembly sequence, inventory management, and factory layout. Once the optimal manufacturing layout has been identified, a manufacturing schedule from this layout can be generated. Figure 2-29 presents the data flow required for developing this simulation.

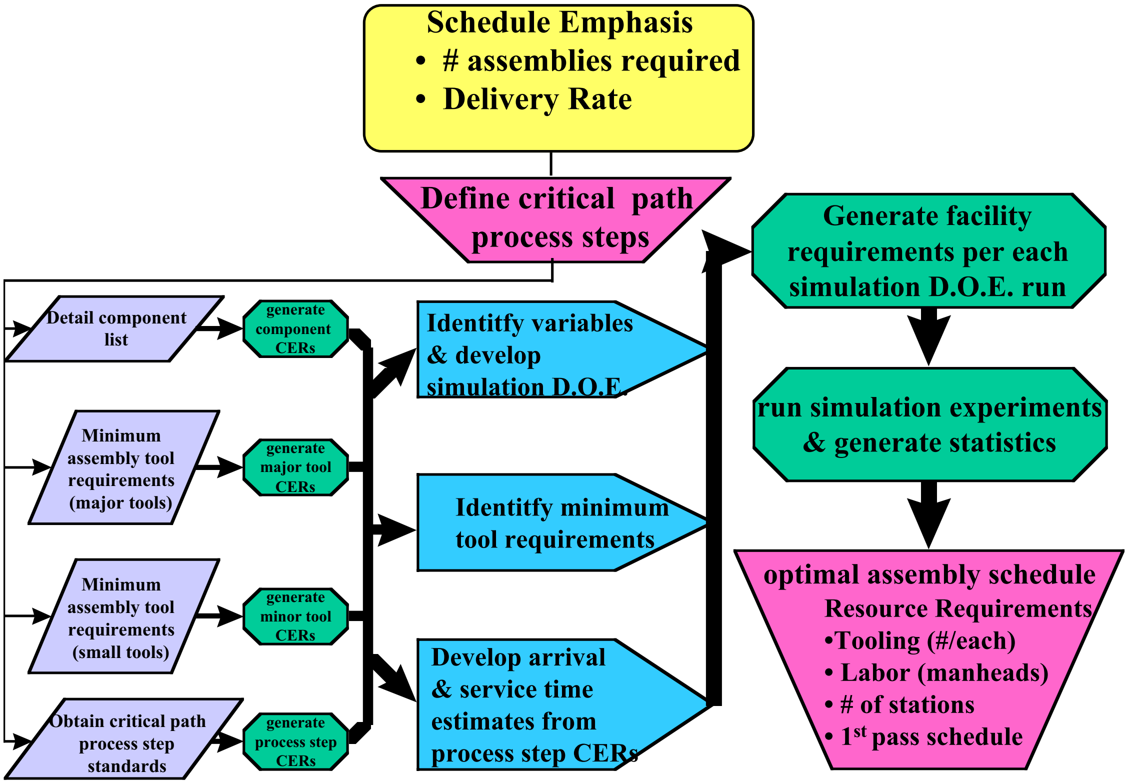

Schedule Emphasis
• # assemblies required
• Delivery Rate
Define critical path process steps
Detail component list
generate component CERs
Minimum assembly tool requirements (major tools)
generate major tool CERs
Minimum assembly tool requirements (small tools)
generate minor tool CERs
Obtain critical path process step standards
generate process step CERs
Identitfy variables & develop simulation D.O.E.
Identitfy minimum tool requirements
Develop arrival & service time estimates from process step CERs
Generate facility requirements per each simulation D.O.E. run
run simulation experiments & generate statistics
optimal assembly schedule
Resource Requirements
•Tooling (#/each)
• Labor (manheads)
• # of stations
• 1st pass schedule

Figure 2-29: Process for Developing Optimal Manufacturing Assembly Schedule

4.5.5 Dimensional Management Analysis

Using the 3D CAD data and the relevant dimension and tolerance data assigned to that CAD model, a dimensional management statistical simulation model is created. This model assesses the tolerance buildup to determine the probability that a part or assembly can be manufactured within the specifications. Over and under constraints are also determined as part of the statistical analysis. The results of these analyses are used to determine a more optimum tolerance or sequencing specification that will improve the overall results of the assembly or manufacturing process.

4.5.6 Risk Assessment

Updated engineering, manufacturing process plan, assembly cell simulation, factory simulation, and the latest schedule data are inputs to SAVE's Risk tool where a revised risk assessment model is generated. The IPPT uses this tool to identify potential risk changes. This early identification of areas of concern allows the IPPT to take positive action to eliminate or reduce the probable hazards. Figure 2-30 presents a schematic of the data required to develop a risk model.

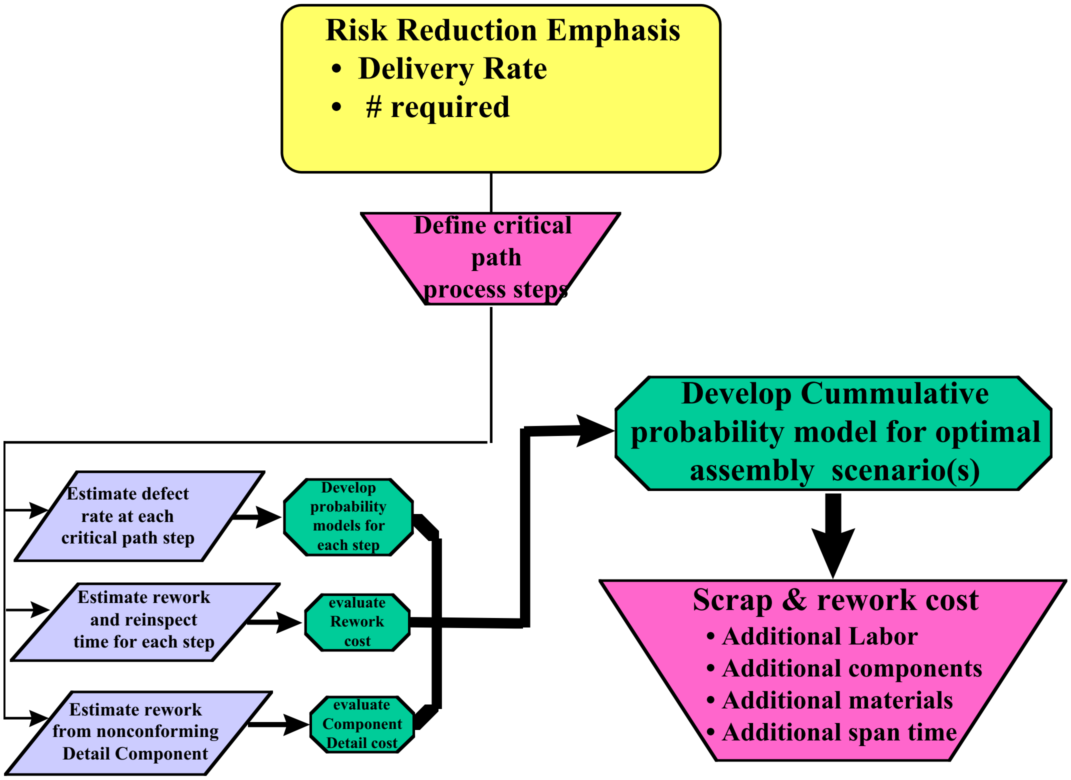

Risk Reduction Emphasis
• Delivery Rate
• # required
Define critical path process steps
Estimate defect rate at each critical path step
Develop probability models for each step
Estimate rework and reinspect time for each step
evaluate Rework cost
Estimate rework from nonconforming Detail Component
evaluate Component Detail cost
Develop Cummulative probability model for optimal assembly scenario(s)
Scrap & rework cost
• Additional Labor
• Additional components
• Additional materials
• Additional span time

Figure 2-30: Data Requirements for Developing a Risk Analysis Model

4.6 Design and Manufacturing Plan

After the release of the preliminary manufacturing plan, the CAD models continue to reflect the latest design. In concert with these changes the manufacturing process plan is reviewed to ensure that it reflects the most current detailed information and models. The factory simulation model is updated to reflect any additional equipment requirements anticipated. Conflicts or collisions in the assembly sequence will be resolved and optimized via the factory, assembly cell and scheduling models. The IPPT will expand their detailed schedules through simulation to optimize span times for internal and external processes. The dimensional management model is updated with the latest sequencing and CAD data to update the risk assessment.

At this point, the optimized design and manufacturing process plan are ready for final review and analysis, in preparation for release. Tool set data, build to print elements including the installation/assembly plan, tool designs, design to cost allocations, factory layouts, and schedules begin to be finalized.

The relationship between the data components and cost, schedule, and risk are reflected Figure 2-31.

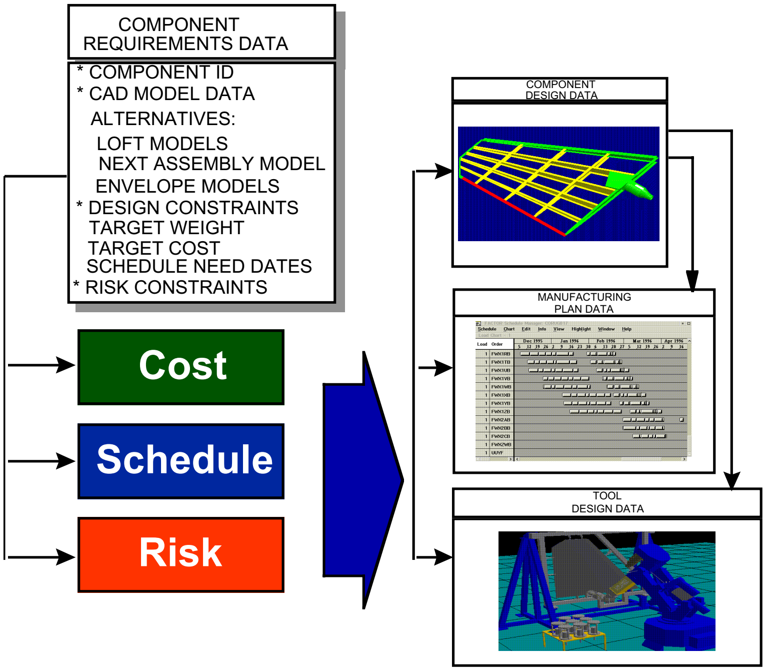


Figure 2-31: Design, Manufacturing, and Tooling Results

4.7 An Integrated EMD Evolution

As the design and manufacturing process plans take shape, the IPPT is able to take advantage of an integrated understanding of the plans, designs and tradeoffs using the SAVE tool suite while fully evaluating the effects of any proposed modifications to the weapon. This development and continuous evaluation process by SAVE, the higher order IPPTs, and the supply chain is repeated and expanded many times as the components merge into complex assemblies and ultimately into total systems. Figure 2-32 shows the interrelationships between design alternative and manufacturing implications.

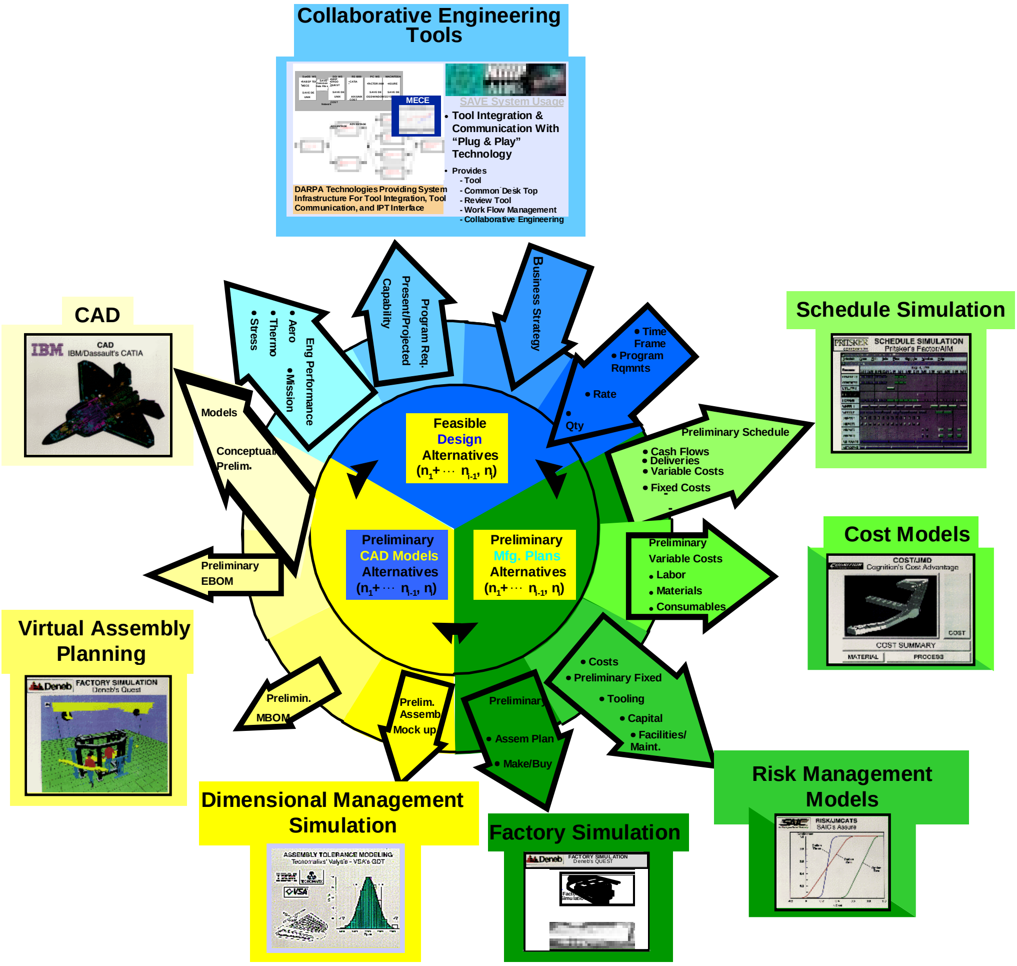


Figure 2-32: Relationships Between Design and Manufacturing

An additional benefit for the EMD management is their ability to utilize the SAVE Tool Suite to evaluate and improve the timing of capital equipment acquisitions/investments, correct logistical problems, determine acceptable limits of materiel and supplier activities.

The SAVE simulation process permits better planning and timing of release of work to the shops, identification of potential trouble spots, and provides "what if" analysis of unexpected occurrences and machine breakdowns. It also provides visualization in the form of animated work instructions once the design reaches the shop floor.

Since SAVE permits IPPT approval of preliminary manufacturing process plans before work begins on more detailed plans, problems and issues can be raised and solved at the preliminary stage before a significant amount of effort has been spent to develop a detailed manufacturing plan that tooling or design does not support.

Chapter 3

Usage Guidelines

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

CHAPTER 3 – TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 Overview

This chapter will describe the general use guidelines of the Simulation Assessment Validation Environment (SAVE) system for manufacturing simulation. This is not intended as a user's manual for the system components (These are provided in Appendices to this document). Rather, is will address the larger issues of when to apply SAVE and how to achieve maximum benefits when applying the environment to a to a design project or trade study.

There are three primary sections in this chapter, each of which addresses an important aspect of using SAVE.

1. Features of the SAVE Environment

2. Guidelines for Effective Use of SAVE Environment

3. Metrics/Benefits Assessment

2.0 Features of the SAVE Environment

The SAVE Virtual Manufacturing Environment features an integrated set of manufacturing simulation tools that are used by a variety of disciplines within the IPPT. The SAVE Data Model, which provides the basis for the integration, is a robust yet flexible representation of the data that is shared among the tools. This section highlights the important features and capabilities that are provided in SAVE.

2.1 Integration

The key to the concept of SAVE is integration. There are a number of vendors who have manufacturing simulation tools on the market today. These tools may be employed independently and produce significant benefits on their own. Their use, however, has typically been quite expensive and therefore quite limited.

One reason for this high cost of operation is the amount of data that is required to develop and run a simulation model. This data is quite time-consuming to gather and input into the different simulation tools. Within the realm of manufacturing simulation and assessment tools, there is a wealth of common or shared data. Figure 3-1 shows some examples of data categories that are common among the classes of tools being addressed within the SAVE Virtual Manufacturing (VM) Environment. SAVE makes the use of these tools more affordable by allowing the IPPT to gather and create the data once while using it many times and in many contexts.

	CAD	Factory Simulation	Assembly Planning	Schedule Simulation	Risk Analysis	Cost Analysis	Variation Analysis
Process Plan / Work Inst		●	●	●	●	●	●
Geometric Models / Defn	●		●		●	●	●
Task Durations		●	●			●	
Resource Estimates		●		●			
Rates and Factors		●	●	●	●	●	
Process Rates		●	●				
Factory Layout / Definition	●	●		●			
Manufacturing Rules				●		●	
Timelines				●			
Feature Definitions	●					●	●
Cost					●	●	
Tolerance Limits					●		●
Risk					●		●

Figure 3-1: Examples of Common Data Among VM Tools

In addition to tool use affordability, integration provides synergistic benefits as well. Although use of the individual tools will identify potential problem areas relative to a specific area or discipline, it does not identify the impact of the problem on other areas of concern or the effect the solution may have on those other areas. When the manufacturing simulation tools are integrated into a Virtual Manufacturing Environment, the IPPT can easily conduct design and process analyses or trades with full knowledge of the relative cost, schedule and risk impacts of their decisions.

2.2 Configuration Management and Control

The SAVE system has been developed to support manufacturing simulations performed during design development. These simulations are performed to assess the manufacturing cost, schedule, and risk impacts of product and process design decisions. The nature of complex product design is inherently iterative and SAVE has been designed to manage the multi-version nature of design simulation data. As a design tool, SAVE-generated data are expected to be released (likely controlled by a PDM) to production and transferred to downstream systems. SAVE provides configuration management of data while it is in work, provides for data storage by a PDM, and allows results to be extracted to downstream systems if the data are not already stored there during development.

The philosophy behind SAVE data management is to provide flexible control that can be tailored by a design team. There is strong capability to create alternative approaches at several levels, lock each alternative as its study is completed, and to denote the selected alternative when the overall design study is concluded. Manufacturing Process Plans are the central data grouping within SAVE. Process Plans may contain sub-Process Plans, allowing a high-level plan to be sub-divided so that the team can easily work on variations of these sub-plans simultaneously. When one version is selected it is easily re-combined into the high-level plan. Alternative approaches may be totally constructed from the ground up, or may be copied from an existing alternative when that is more efficient. Low level versioning of individual data attributes is provided to minimize the need to create a new version when only small changes are made. These versioned variables maintain a history of their values, and information on when and how they were changed. All data objects keep track of the number of times they are referenced in other objects, and most user-initiated "deletions" are simply removal of a reference and a decrement in the reference count. When references to non-library objects reduce to zero, they are actually deleted from the database. Objects in SAVE libraries can only be deleted when all other references have been removed. See the SAVE Data Model Usage Guide section of this report for a complete description of the capabilities discussed above.

SAVE users must develop an understanding of SAVE's Data Model and the data configuration management capabilities it provides. This understanding will allow a team to quickly identify the paths to be included in a design study and the best representation of the data within SAVE.

The elements of SAVE configuration management include:

- Status Flags on several key data elements - lower level data controlled automatically
- Alternatives supported for Design Studies and Process Plans
- Copy Command - intelligent copy of Process Plan to start alternatives
- Remove/Delete - Tracks references to data objects by other objects
- Versioned Variables - Minimizes need to create alternatives
- Back End Data Storage - Data management of physical storage system

Each of these elements and their use are discussed below.

2.2.1 Status Flags

A Status Flag data attribute is included in the Design Study, Design Alternative, and Process Plan data objects within the SAVE Data Model. These flags are used internally, and can be viewed using the Query Manager. These flags can take on values of:

1. Working
2. Review
3. Released

Data can modified in these objects only when the Status Flag is set to Working. Both the Review and Released status lock the data object from change. The Review status denotes that the data are in the review and approval cycle while the Released status denotes that the approval cycle has been completed. Users with appropriate authority can change the value of flag from a higher level to a lower level if it becomes necessary. This is in keeping with the fact that SAVE is a design study tool and when a study is complete, the data from the selected alternative is expected to be formally released through a PDM and passed to a downstream system. Status flags are set using the Query Manager or other user interface into the SAVE data model.

While essentially all data within SAVE have a status flag, only the three objects listed above expose their status flags to users for control. The status flags on lower level data objects are controlled by the objects in which they are referenced. For example, When a Process Plan status is set to "Released" all of its associated Operations are also locked. Likewise, the Cost data object referenced by an Operation is also locked. Some objects that are used in multiple Process Plans (for example Reference Processes) are not locked by a given Process Plan. Many Library objects are created specifically to be used repeatedly and should not be modified after their initial creation. If a modification is needed a separate version of the library object should be created and clearly noted in its Name and Description attribute fields.

2.2.2 Alternatives

The SAVE Data Model provides two levels of alternatives to be identified and included in a Design Study.

1. Multiple Design Alternatives in a Design Study

2. Multiple Process Plans in a Design Alternative

A Design Study may involve only a single approach to be studied and its cost, schedule, and risk assessed. But in many cases a team will identify two or more alternative approaches to a design problem and SAVE is an ideal tool to use in evaluating and selecting the best alternative. Each of these first-level alternatives should be designated as a separate Design Alternative referenced by the Design Study.

A second level of alternatives is provided by the capability for a Design Alternative to maintain a list of possible Process Plans. This capability is included to allow a design team to consider one or more subtle variations on the basic Design Alternative. Use of these alternatives is not rigidly forced by the SAVE system and it is up to the design team to plan their usage. Creating a Design Alternative is a very simple task and is performed by interactive access to the Data Model. Each alternative Process Plan can be created from scratch, or an alternative can be created by copying an existing Process Plan and making the desired changes as simulations are executed. This copy process will typically be performed by one of the design team members using the interactive Data Model access tool, currently the SAVE Query Manager. The access tool also provides the capability to view a list of design alternatives as well as the process plans that are part of that alternative.

In deciding what alternatives to create and use, the team must remember that there is no facility for merging two divergent plans, even if they were created as copies of the same plan. This is no different than having a team of people create a single word processing document. Some manual version coordination is required. Two alternatives may be created and studied, but ultimately one will have to be chosen as the final version.

Users should also understand and use the nested process plan capability within SAVE as it impacts alternatives. One Operation within SAVE can, in fact, be a sub-Process Plan. This nesting capability was originally included to support a high-level/low-level abstraction within Process Plans allowing some tools to simulate a higher level view while other tools simulate at a lower level. Plans can be nested to any depth, but typically only one to three levels are needed. Using this nesting feature, sub-process plans can be individually modified and a high-level process plan can be constructed to combine the best sub-plans. This is analogous to a team working on chapters of a report rather than working on multiple copies of the full report. A little creative thinking allows a very flexible capability.

2.2.3 Copy Command

An ability to easily copy some objects is provided to simplify the creation of alternatives. All objects that reside in SAVE libraries may be copied.

The copy capability is intelligent in that copying a data object automatically makes copies of all appropriate included objects. For example, copying a Process Plan automatically makes a set of Operations that are copies of the original set, and as one of these Operations is copied, its CostInfo, RiskInfo, and ScheduleInfo objects are copied and the copies associated with the new Operation. The copy command also has the capability to reference data objects that should be used instead of copied. Using the Process Plan example above, the reference process objects associated with each operation would be referenced, not copied.

2.2.4 Remove / Delete

Each data object within the SAVE system keeps track of the number of times that it has been referenced by other data objects. In an object-oriented system such as SAVE, when an object is not referenced by another object, access to it is totally lost and it can (and should) be deleted. Data objects that need flexible long-term access, independent of other use within SAVE, are maintained in Libraries (just a special data object themselves).

Data objects track their own usage, totally freeing the user of this burden. Users are free to create objects and add or remove them from being referenced by other data objects. Therefore, all user "deletion" actions are actually reference removals. When non-library objects have their last reference removed, they are automatically deleted. A user can remove all references to a Library object, but the object remains because it is in the library. A user may remove a data object from the library, and if no other references exist, the object is deleted. Conversely, a user may remove a data object from the library (for example, remove a Reference Process that has been superceded) and it will not be visible there, but it will still exist if it is referenced in one or more Operation. As a point of information, deletion is actually performed the next time the server is shut down.

The SAVE data object removal / deletion scheme assures that no necessary data are ever lost and provides a scheme that is logical and simple for users to follow.

2.2.5 Versioned Variables

By the very nature of SAVE being a system for design studies, its data will be continually added to and modified. To maintain a record of this evolution would be prohibitive if new versions of the higher-level data object were needed every time one low-level value changed. For instance, it would not be practical to have a new version of a Process Plan simply to track the fact that one cost element of one Operation had been updated.

SAVE has implemented a novel approach to this potential problem. Many of the rapidly changing low-level values are stored in Versioned Variables (Floats and Strings). These Versioned Variables are data objects themselves and they maintain the history of all values that a variable has held. Each value is stored along with a date-time stamp and a record of the tool that generated the value. By default, only the most current value is returned on simple queries, but any previous value (by date) or the entire history can be accessed if needed. If needed, the status of the a Process Plan on April 1 at 9:30 AM can be determined. The implementation details of Versioned Variables are discussed in the Data Model Users Guide section of this report.

2.2.6 Back End Data Storage

The SAVE Data Model has been implemented to present a consistent view of stored data to SAVE-compliant client software and yet to allow the actual data storage to be distributed to multiple physical back end data stores. Most organizations that install SAVE will already have manufacturing process data stored in some electronic systems. SAVE was designed to minimize the requirement to replicate that data with its associated configuration issues. The SAVE Data Model server can read and write data to the existing database and still provide the standardized access to these data to client software tools.

The way that a SAVE server provides this distributed storage capability will likely vary among different commercial servers. During implementation users will identify which data objects and attributes they want to have physically stored in existing systems. This information will be fed to the server as data, and should require little or no re-coding of the server software.

In many installations today, one of these backend data stores will be a Product Data Management (PDM) system. With minimal tailoring, a server can be extended to make calls to the PDM Application Protocol Interface (API) or CORBA interface. In this way the key data accessed from a SAVE server can be managed consistently and as formally as other product design data.

2.3 Interdisciplinary Interactions

Implementation of an integrated virtual manufacturing system such as SAVE can face significant cultural barriers. Not the least of these is the fact that, in many companies, the integration crosses a wide range of organizational boundaries. Even with its current, modest level of integration, SAVE supports tight interaction among Project Engineering (design, tolerance analysis), Manufacturing (manufacturing processes, scheduling), Business Operations (cost), Systems Engineering (risk), and Industrial Engineering (factory layout), as an example. Old barriers, particularly not wanting to share in-work data, must be broken down.

Each discipline must recognize that they are an integral (but not dominant) part of a team. The heart of the SAVE system is a logically central database containing simulation information. It is owned and managed by the team. This database is explicitly designed to support iterative refinement of data from a number of sometimes-conflicting sources. As simulations are run these data are refined until they achieve a maturity that supports their use by the team in design trades. Sharing in-work data is a necessity and the team must accept this and must understand and communicate the accuracy of the data as it evolves. The collaborative design notebook is a key tool for this communication.

Full use of an integrated manufacturing simulation environment can involve a moderate number of disciplines within a design team. The final selection of team members, and the tools to be used often depends on the design problem to be addressed. The SAVE system has been developed to be very flexible, providing capability for the most difficult problems, but not forcing additional work on smaller problems. Not all design decisions require detailed analysis of cost, schedule, and risk. Any combination can be called for and it is up to the team to make these choices.

However, the team must fully understand the tools at hand, for many of the capabilities are complementary or overlapping. For example, a cost tool may have an estimate of schedule information that is appropriate for certain steps of the estimate refinement process. The cost tool may need to be run even if it is not the primary driver. In the past, the full range of tools was not brought to bear on problems due the isolation of disciplines working on a design. The integration within SAVE can help a team view the overall problem as one of jointly refining all necessary decision support information.

The current SAVE system includes seven classes of simulation tools, often implying six separate disciplines or organizations. These classes include:

1. CAD
2. Assembly Variability Simulation
3. Assembly Simulation
4. Factory Simulation
5. Schedule (discrete event) Simulation
6. Cost Modeling
7. Risk Assessment

The matrix in Table 3-1 shows, at a high level, the interactions of these disciplines in terms of the data that may flow among them.

The role of each discipline is to produce results that support a design decision, and to provide information to other disciplines that they need to accomplish their jobs. This second task is too often relegated to a lower priority. A team environment, supported by an integrated toolset, can develop a much better balance of these tasks especially when the product information shared among the tools is automatically made available via the SAVE integration.

Table 3-1: Engineering and Manufacturing Discipline Utilization of SAVE

SAVE TOOL USAGE TOOL DISCIPLINE	CAD	FACTORY SIM..	ASSEMBLY CELL SIM	ERGONOMICS	SCHEDULE SIM.	COST MODELING	RISK MODELING	DIMENSIONAL MGT
STRUCTURAL DESIGN	●		●	●		●		●
MECHANICAL DESIGN	●		●	●		●		●
ELECTRICAL ENGINEERING	●		●	●				
TOOL DESIGN	●		●	●				●
NC PROGRAMMING	●							
FACILITIES ENGINEERING	●	●	●					
PRODUCIBILITY ENGINEERING	●		●	●	●	●	●	●
QUALITY ENGINEERING	●		●	●	●	●	●	●
SUPPORTABILITY ANALYSIS	●		●		●			
INDUSTRIAL ENGINEERING		●	●	●		●	●	●
PLANNING		●	●	●				●
MASTER SCHEDULING					●			
FINANCE						●		
VALUE ENGINEERING						●		
MANUFACTURING ENGINEERING	●	●	●	●				●
IPPT LEAD		●	●		●	●	●	
PROGRAM MANAGER		●			●	●	●	

2.4 SAVE Data Model

The heart of the SAVE solution is the SAVE Data Model (SDM), described in this section. While the other elements of the SAVE system (Work Flow Manager, Query Manager, and Collaborative Notebook) are important, it is the SAVE Data Model that provides the basis for integration among the commercial simulation codes.

It is important to understand that the SDM does not attempt to supplant a Product Data Management (PDM) system, nor to manage all of the data used or generated by a suite of manufacturing simulation codes. Rather, the SDM is designed to contain and provide access to the information that is shared among a suite of codes. Each code is expected to output to the SDM any data that might be used by other codes or by the design team to make design decisions and to input data from the model that it needs to perform its own simulations.

The tremendous range of capability of commercial simulation codes demands a very flexible data sharing environment and has strongly influenced the design of the SDM. The SDM, in fact the whole SAVE system, does not attempt to prescribe rigid operating constraints for which tools to use, the order in which they are executed, or the data that each tool shares through the SDM. These decisions are left to the design team and can be varied to suit a particular design study. This flexibility requires that developers of the simulation scripts and other models (that are executed by the commercial tools) be very familiar with the data within the SDM, as they must establish and use the naming conventions that are discussed below. The persons who finally execute the simulations must also understand the SDM, but to a lesser degree, as they will control the use of any input/output options provided by the simulation codes in their interfaces to the SDM.

A key approach taken by SAVE is to provide a truly open architecture which allows a plug-and-play integration of many tools. Specifications for the SDM have been widely distributed during development to help achieve widespread acceptance and maximum flexibility. This non-proprietary approach allows any tool vendor to provide a SAVE-compliant version of their tool that can be easily incorporated into a SAVE system without further integration required from users. SAVE uses a central repository approach for the SDM as opposed to tool-to-tool integration. This approach best fits the nature of the supported codes, which tend to be highly interactive and can require extended execution sessions.

2.4.1 Development of SAVE Data Model

The initial version of the SDM was developed by a group of end-users and system developers in a top-down manner, and did not evolve from a detailed list of inputs and outputs from a set of simulation tools. The intent of this group was to model the information that a design team would need to develop in order to support product and process design decisions based on manufacturing considerations of cost, schedule, and risk. It was anticipated that this model would be highly representative of the data that is used in current design activities in order to minimize the learning curve for the use of the SAVE system. Further, although the model was originally developed in an aerospace community, every attempt was made to generalize the model to make it applicable to any mechanical design problem. At every stage of development, the model was shared with as large an audience as possible, through presentations, distribution of documents, and by openly publishing the specification on the SAVE web site. Anyone willing to spend the time was encouraged to review the model, ask questions, and suggest changes. Many of the features of the current model originated from outside the SAVE development team, and have significantly strengthened the model.

After several rounds of model development, the SAVE team mapped the input and output variables from the set of SAVE simulation tools. Virtually no changes were required in the model to accommodate these tools. The current SAVE tools cover a wide range of classes of manufacturing simulation and the team is confident that the model will support most tools within this problem domain.

2.4.2 The Object-Oriented Nature of the SAVE Data Model

The SDM uses an object-oriented approach for representing the information shared among simulation tools. Software objects are ideally suited for representing complex data. First, an object contains not only data values, but also can contain active methods to perform operations on this data, significantly enriching the ability to represent complex information and make it easily accessible in software systems. Second, complex relationships are expressed in a very natural manner, and access to related data can be very efficient. Data values within one object can, in fact, be other data objects. Once the basic concept is understood, complex data relationships can be easily mastered. Data objects are easily accessed without the need to retrieve small pieces and rebuild the complete objects.

Many early SAVE users had experience with relational databases, and seemed to try to understand the SDM in terms of a relational model. They asked questions like, "Once I have a process plan, what key fields do I need to access the individual operations?" Some attempts were made to relate the data model to a set of spreadsheets. Along the way the team developed a set of hyperlinked web pages to document the model for user training and were pleasantly surprised to find that the linked pages operate logically very much like the object-oriented data model itself. Each data object is a page that contains textual descriptions of the object, its data fields, and any methods that it can perform. Where a data field is, in fact, another data object and not simply a string of text or a number, it is hyperlinked to that data object's web page. The hyperlink acts just link the object linking pointer and for all practical purposes the second data object is a part of the first data object. When users started using the web page data model description, they rapidly understood the model. These web pages are accessible on the SAVE website at http://skipper.mar.external.lmco.com/save and are reproduced in Appendix C of this document.

2.4.3 Key Groups of SAVE Data Objects

Data objects within SAVE fall into six categories:

1. Core Process Plan
2. Resources
3. Part Information
4. Results
5. Model Management
6. Utility Objects

The manufacturing process plan and its constituent operations or job steps are the central elements of the SDM. They provide the basic structure for information generated and used by the simulation tools. Data objects in this group include:

1. Process Plan
2. Operation
3. Reference Process
4. Manufacturing Order

Resources define the personnel and tools that are needed to perform the operations within a process plan and are central to defining schedules and cost. The SDM supports identifying necessary resources and tracking their use. The related data objects include:

1. Resource
2. Personnel
3. Work Calendar
4. Work Shift
5. Break
6. Tool
7. Breakdown
8. Resource Pool
9. Resource Application

Information about the parts being manufactured is another area of the model. The part feature information is one area that many people are interested in expanding to allow the model to support problem domains other than manufacturing simulation. The data objects in this group include:

1. Part
2. Part Usage
3. Part Location
4. Feature
5. Material
6. CAD Model

Result information within the SAVE system includes estimated cost, schedule and risk. It is these data elements that collect the initial and refined estimates that are calculated by various simulation tools and are ultimately the basis for design decisions among alternative design approaches. The data objects in this group are:

1. Cost
2. Schedule
3. Risk

Several data objects have been defined to help teams organize and manage the many process plans that will be stored in the SDM. These data management objects include:

1. Design Study
2. Design Alternative
3. Manufacturing Program
4. Simulation Request

A number of supporting data objects are defined to support the groups listed above. In general, these objects work behind the scenes and while users should be aware they exist they will not need to known their details. They are included here for completeness.

1. Base Object
2. Characteristic
3. Contributor
4. DbAccess

5. Date-Time
6. Inflation Table
7. Library
8. Named Object
9. Object Sequence
10. Simulation Model
11. Value With Units
12. Versioned Float
13. Versioned String

2.4.4 Discussion of Data Object Use

Process Plan

The process plan is the central data object in the SDM. It contains a sequence, or list, of operations that define the steps needed to complete some portion of a part or assembly. Other data fields contain information on available personnel and tool resources and the lot sizes to be produced by the plan. Some detailed schedule information is included, as are the rolled up cost, schedule, and risk that are detailed in each operation. In order to maintain configuration control, a status flag is available that can lock the data elements associated with a process plan when that plan is released. In addition, a list of simulation tool executions that were used to develop the information about this process plan is maintained in a sequence of SimMod data objects.

Like many SDM data objects, the process plan includes a list of Characteristics which are simply Name-Value pairs that can be used to dynamically expand the list of data associated with a process plan.

Every process plan that is created is stored in a library to make it easily accessible.

Process plans and their related objects can be modified and expanded as a design study progresses. One design alternative (discussed below) can have several alternative process plans that may be explored before one is selected as the preferred or baseline. There is no provision to merge two alternatives of a process plan that have been concurrently modified in different areas. This is the classic problem of two people modifying two separate copies of a document and wanting to merge the changes. In general, this is a difficult (if not impossible) task. One of the alternatives would have to be selected as baseline, and the other changes made by executing simulation codes to make the changes. The SDM does have one feature, nesting of process plans, which can help alleviate this problem, if used correctly. Any operation within a process plan can, itself, be another process plan (see the description of operations). If appropriate, two users can concurrently modify two of these sub-process plans and are secure from causing overlapping changes. The sub-plans are easy to reference and changes are inherently reflected in the top-level plan. The SDM's nested process plan capability can be used in other ways and adds great flexibility to the model's versatility.

Operation

An operation models one step in a process plan. As mentioned above, an operation can be made a sub-process plan by simply associating the sub-plan with the operation's ProcPlan attribute. An operation contains attributes detailing certain timing information in addition to schedule attributes. Basic cost, schedule and risk are recorded in attributes that contain data objects of those types.

An operation may contain a reference process that has general information that pertains to all occurrences of that type of operation. For example, the standard hours for an operation can be stored in the reference process and are available to a particular operation by declaring it to be of that type.

Defining additional attributes in a sequence of characteristics can dynamically expand the operation’s data objects. The characteristics list for an operation is automatically updated from the reference process when it is declared, so that all operations of a given type have the same list of characteristics.

Simulation tools identify an operation from its name attribute. The ID and Type attributes can supplement this information.

Resources used by an operation are recorded in two lists, PersonResApplic and ToolResApplic. These resource applications refer to an available pool of resources associated with the process plan and are used to record the amount of a given resource utilized by this operation.

Features associated with an operation are contained in lists of these data objects. Parts are associated with the operation through a series of part usages. They are contained in either the consumed parts or the produced parts list.

The ordering of operations within a process plan is recorded in a sequence of precedent operations contained within each operation. In this way, each operation knows a list of all operations that must be completed prior to the initiation of this operation.

Reference Process

The reference process data object contains information that is applicable to a general class of operations and may be needed by each occurrence of that process. Standard hours or Cpk are examples of this sort of information. A list of characteristics is provided to dynamically expand the information associated with a reference process. A second list of characteristics, called OpChar, is included and this is the list that will be automatically added to an operation's characteristic list when it associated with a reference process.

Manufacturing Order

A series of manufacturing order data objects is associated with each process plan to identify the quantity and schedule for each batch of parts that will be produced.

Resource

Resources, in particular tracking their utilization, is probably the most complex portion of the SDM. The resource data object itself is simply a parent class from which both personnel and tools inherit some common attributes (efficiency and associated CAD model).

A short discussion of the use of resources in the SDM is appropriate here. There are two basic types of resources, personnel and tools. They and their related data objects will be discussed in more detail below. Tracking resource availability and utilization is a key element of accurate manufacturing simulation. Determining necessary resources and avoiding bottlenecks are a major reason for performing simulation and identifying these costs is the central element of accurate cost estimations.

Many types of personnel and tool resources can be created within the SDM. They are stored in a library to allow their general use within many process plans. A resource pool data object represents the specific instance of the availability of a resource in a process plan. Process plans may contain many personnel and many tool pools. Each pool is uniquely related to one process plan and identifies the underlying resource and the quantity available. As resources are used by operations within the process plan, the pool records the total utilization against the available quantity.

Any operation that needs a resource must "acquire" it by creating a resource application data object that references a particular pool associated with the process plan in which the operation exists. This resource utilization data object is uniquely associated with one operation and is used to track the amount of the resource utilized by that operation.

Personnel

Personnel resources obviously represent people and expand the attributes of a resource to include information like skill, labor rate, work calendar, and work shift information. Personnel are made available to process plans and used in operations as described in the Resource section above. Personnel can have an associated CAD model to allow detailed representation in a simulation.

Work Calendar

Work calendar data objects contain the information on the high level availability of personnel. Methods are included in this object to properly define the available work days, to check if a personnel is available on a given day, and to easily determine the number of available days between two dates.

Work Shift

The work shift data object defines the available working hours for personnel, including any breaks within the shift.

Break

The break data object allows definition of multiple breaks, including such work stoppages as lunch and dinner breaks, that will be associated with a given work shift. The object includes break start and stop times.

Tool

Tool data objects are used to represent all classes of resource tools from large machine tools, to hand tools, including project specific jigs and fixtures. Tools are made available to process plans and used in operations as described in the Resource section above. Tools can have an associated CAD model to allow detailed representation in a simulation.

Tools maintain information about their cost, tolerance capacity, and failure rate. They contain a sequence of characteristic data objects to allow dynamic expansion of their data attributes.

Tools also contain a list of the breakdown data objects to held define their availability, and the impact of repairs.

Breakdown

Breakdown data objects define the likelihood (hours to first failure and mean time between failure) and consequences of breakdown. The consequences can be calculated by a simulation tool by using the time to repair and required repair resource information.

Resource Pool

An overview of resource utilization is given in the Resource section above. A resource pool data object represents the specific instance of the availability of a resource in a process plan. Process plans may contain many personnel and tool pools. Each pool is uniquely related to one process plan and identifies the underlying resource and the quantity available. As resources are used by operations within the process plan, the pool records the total utilization against the available quantity. Changes to a resource utilization should only be made by using the SetQuanity method within the Resource Application data object associated with each operation.

Resource Application

An overview of resource utilization is given in the Resource section above. Any operation that needs a resource must "acquire" it by creating a resource application data object that references a particular pool that is associated with the process plan in which the operation exists. This resource utilization data object is uniquely associated with one operation and is used to track the amount of resource utilized by that operation. Changes to a resource utilization should only be made by using the SetQuanity method within the Resource Application data object associated with each operation.

Every resource application data object contains a unique transformation matrix to allow the resource to be positioned within a simulation.

Part

This data object defines the part being manufactured. For simplicity, it may be a detailed part or an assembly. Assemblies are a part that contains a list of associated parts. Part data objects directly contain some attributes and are further expanded by having other associated data objects including cost, CAD model, material, associated parts, and features.

Parts are referenced in both process plans and operations.

The SDM contains the concept of both a Manufacturing and Engineering Bill of Materials (BOM). This BOM is represented by a part object that contains a list of associated part objects. The process plan object contains both a MBOM and an EBOM that is essentially this indentured parts list. The EBOM may be populated from the CAD data file and used as a starting point for the manufacturing simulations. The MBOM will typically be defined as a result of performing simulations.

Part Usage

The data object defines the quantity of a particular part that is used in a given operation. It also contains a list of part locations for the given part.

The name of a part usage is the same as the part number in the part data object.

Part Usage can represent both consumed and produced parts as associated with an operation.

Part Location

Part Location defines the transformation matrix for each part that is used in an operation. This matrix is used by the simulation codes to position the part correctly during model generation. The number of part locations in a part usage is equal to the quantity attribute in the part usage.

Feature

Features may be either design or manufacturing features depending on their use. The feature data object is quite simple at this time, mainly being a list of characteristics (Name-Value pairs) that can be dynamically populated by users. Type, quantity, and cost information are explicitly included.

Growth of the SDM in the design feature area is one of the most exciting near term possibilities. The need to provide widely accepted definitions of intelligent design information in a readily accessible form to a wide range of CAx tools is a key driver for acceptance of the SAVE approach and rapid expansion beyond just manufacturing simulation.

Material

An easily expandable material data object is included in the SDM. All materials that are defined are stored and made available for any use from a library within the SDM. The standard data attributes of material (type, form, and unit cost) can be easily expanded using the list of characteristics.

CAD Model

CAD models are not explicitly stored in the SDM, but rather are pointed to by the CAD model data objects. The CAD model objects record the format and storage location and include a definition of the bounding box to denote the model physical size.

Cost

The cost data object contains the basic set of cost data used throughout the model. Not all of the cost object attributes will be utilized in every instance of its use.

The fiscal year for which the costs are estimated is included and an inflation table is also included to allow the cost to be adjusted to any year.

All data attributes within the cost data object are stored as versioned floats. Versioned Floats are data objects that maintain a historical list of all values that have been assigned to this variable as a design study progresses. The most recent value is returned by default, but earlier values can be obtained by date or as a complete list. This low level versioning of the data in the SDM significantly simplifies the configuration management of a design study and eliminates the need to version top level data objects when just one value is updated.

Schedule

The schedule data object contains the basic set of schedule data used throughout the model. Schedule data objects include both planned and actual schedule information. Not all of the schedule object attributes will be utilized in every instance of its use.

All data attributes within the schedule data object are stored as versioned variables (float and string). Versioned variables are a data object that maintains a historical list of all values that have been assigned to this variable as a design study progresses. The most recent value is returned by default, but earlier values can be obtained by date or as a complete list. This low level versioning of the data in the SDM significantly simplifies the configuration management of a design study and eliminates the need to version top level data objects when just one value is updated.

Risk

The risk data object contains the basic set of risk data used throughout the model. Risk data objects include both probability and consequence of failure, mean and standard deviation, and Cp and Cpk data. A list of contributors is included to help identify efforts to reduce risk. Not all of the risk object attributes will be utilized in every instance of its use.

Some data attributes within the risk data object are stored as versioned floats. Versioned floats are a data object that maintains a historical list of all values that have been assigned to this variable as a design study progresses. The most recent value is returned by default, but earlier values can be obtained by date or as a complete list. This low level versioning of the data in the SDM significantly simplifies the configuration management of a design study and eliminates the need to version top level data objects when just one value is updated.

Design Study

On any large design project using SAVE there will be a large number of process plans defined in the SDM. To organize these process plans and assist in the overall data management, several high level data objects are defined to group and provide logical access to lower level data. The design study data object is the "top" level of the SDM. It contains the data that define the overall problem that is being addressed through execution of a set of manufacturing simulation tools. The list of alternatives being considered is recorded here and when a decision is made, the selected alternative is also noted. A top-level configuration management status flag is included to denote and control the status of the data within this design study,

Design Alternative

Within a design study there can be many design alternatives. Each alternative data object has a list of alternative process plans that are being considered, with one being denoted as the baseline. A manufacturing program data object is included to contain the overall program information that is of importance to the design study. The manufacturing program data is associated with the design alternative to allow the impact of optional program parameters to be investigated. A configuration management status flag is included to denote and control the status of the data within this design alternative.

Manufacturing Program

The manufacturing program data object contains information on the production quantity and production rate of the program. Several top-level parameters that can drive the program schedule are included.

Simulation Request

This data object seems to cause some confusion. It is simply a means to pass all required start-up information to a simulation code when it is executed. This information may include input filenames, input and output options, and always includes references to the process plan, design study and design alternative that are to be simulated during this execution of the code. The design team creates simulation requests as they create the work flow for the design study. The work flow defines order in which simulation tools are to be run. Each tool execution is associated with a particular process plan by passing a reference to that process plan in the simulation request data object.

This object contains the primary entry points into the data model that might be needed by a simulation code. It is expected that this will be passed to the simulators by the Work Flow Manager. The attributes InputFiles, InputOptions and OutputOptions can be used to pass a simulator information on launch, input, and output options. These attributes use tab-delimited strings to separate the options.

Typically the design team will create these SimReqst objects and include a reference to them in the work flow being developed. The Work Flow Manager passes this reference to the simulator before launch. The simulator wrapper then accesses the SimReqst object to obtain launch options and database context information. Each simulator will determine the syntax of information in these strings.

Base Object

The base object is included in the SDM to contain information and/or methods that are needed by all other objects. Users do not need to be concerned with this object.

Characteristic

Characteristic data objects are used extensively throughout the SDM to dynamically expand data objects to include user-defined attributes. In all cases characteristics are used in lists. A single characteristic data object includes a textual value (string) and a numeric value that is actually a value with units data object (Name-Value pair).

The use of characteristics must be carefully coordinated within the IPPT as well as between the IPPT and the tool vendors. In order for characteristics to be utilized, the vendor codes must be aware of their existence and must update their tool wrappers to accommodate the characteristic. Some tools allow user mapping of variables. In these cases, no software modifications would be necessary.

Contributor

Contributor data objects are used in risk objects to define the source of risk. Each contributor records the name and description of the risk and its percent contribution to the total risk.

DbAccess

The DbAccess object provides the methods that control client access to the SDM server. These methods include basic connection capability, transaction control, object creation for named objects, and access to Libraries and SimRequests.

Date_Time

Date_Time provides a standard time stamping data object that is used for every object when it is created. Standard date-time format is yyyy/mm/dd 24:mm:ss.

Inflation Table

This data object contains a table of inflation values for a range of years. Many inflation tables may be defined and they are stored in one of the SDM libraries. Each table includes a rate that will be used to extrapolate beyond the listed years. Inflation values are stored values relative to some year rather than discrete values for that year.

Library

Library data objects are lists that contain all objects of a given type that have been created in an SDM. The actual list in a library object is a SDM Sequence object which provides methods to add, find, and retrieve objects stored in the library.

The libraries that are defined in the SDM include:

- Break
- CAD Model
- Design Alternative
- Design Study
- Inflation Table
- Material
- Manufacturing Program
- Part
- Personnel
- Process Plan
- Reference Process
- Simulation Request
- Tool
- Work Calendar
- Work Shift

Applications that interface with the SAVE data model automatically place objects in the appropriate library as they are created. This happens behind the scenes with no intervention from the user. In the same manner, when a user creates an object using the Query Manager or other interactive data model access program, that program handles putting the object in the appropriate library.

Named Object

The named object data object is a low-level object that contains attributes that are common to all SDM objects that have explicit names. Named objects include a description and a date-time stamp.

Object Sequence

The object sequence data object provides a standard capability to handle lists within the SDM. The number of items currently in the list is easily available. Attributes and methods are provided to insert, add, delete, define the object type, find by name or index, and return the entire list (with either just data objects or objects with names and descriptions).

Simulation Model

Simulation Model data objects store information about the simulations that are performed for a specific alternative process plan. Information recorded includes the tool name, simulation class, date-time stamp, and the location of data sets used in the simulation.

This object is intended to record an execution of a simulation tool after it is run, not to pass information to a tool as it is run. The SimReqst object is used to provide runtime information to a tool.

Value With Units

The value with units data object is primarily used in the characteristics data objects which allow users to dynamically expand the attributes in many SDM data objects. Value with units implements a Name-Value pair - actually a set of name-value-units information. The units information allows the value with units objects to provide methods to automatically convert the stored value to any other consistent units requested by a user or simulation code.

Versioned Float

Versioned float data objects implement fine-grained versioning for variables in several SDM data objects. As new values are estimated or refined for a stored attribute, old values are retained on a dated list rather than being overwritten. The latest value is returned by default, but earlier values can be recovered by date, or the whole history list may be obtained. Requesting a value by date simply returns the latest value that matches or predates the specified date-time.

This low level versioning of the data in the SDM significantly simplifies the configuration management of a design study and eliminates the need to version top level data objects when just one value is updated.

Versioned String

Versioned string data objects implement fine-grained versioning for variables in several SDM data objects. As new values are estimated or refined for a stored attribute, old values are retained on a dated list rather than being overwritten. The latest value is returned by default, but earlier values can be recovered by date, or the whole history list may be obtained. Requesting a value by date simply returns the latest value that matches or predates the specified date-time.

This low level versioning of the data in the SDM significantly simplifies the configuration management of a design study and eliminates the need to version top level data objects when just one value is updated.

3.0 Guidelines for Effective Use of SAVE Environment

A team employing SAVE should read and understand the guidelines in this section in order to get the most out of the Virtual Manufacturing Environment.

3.1 Planning Activities Prior to the Use of SAVE

Effective use of the SAVE environment requires that design teams perform some initial planning activities prior to using the toolsuite. This planning will not only outline the trade studies themselves but will also identify the areas where simulation may provide the most benefits. Figure 3-2 shows the recommend planning activities.

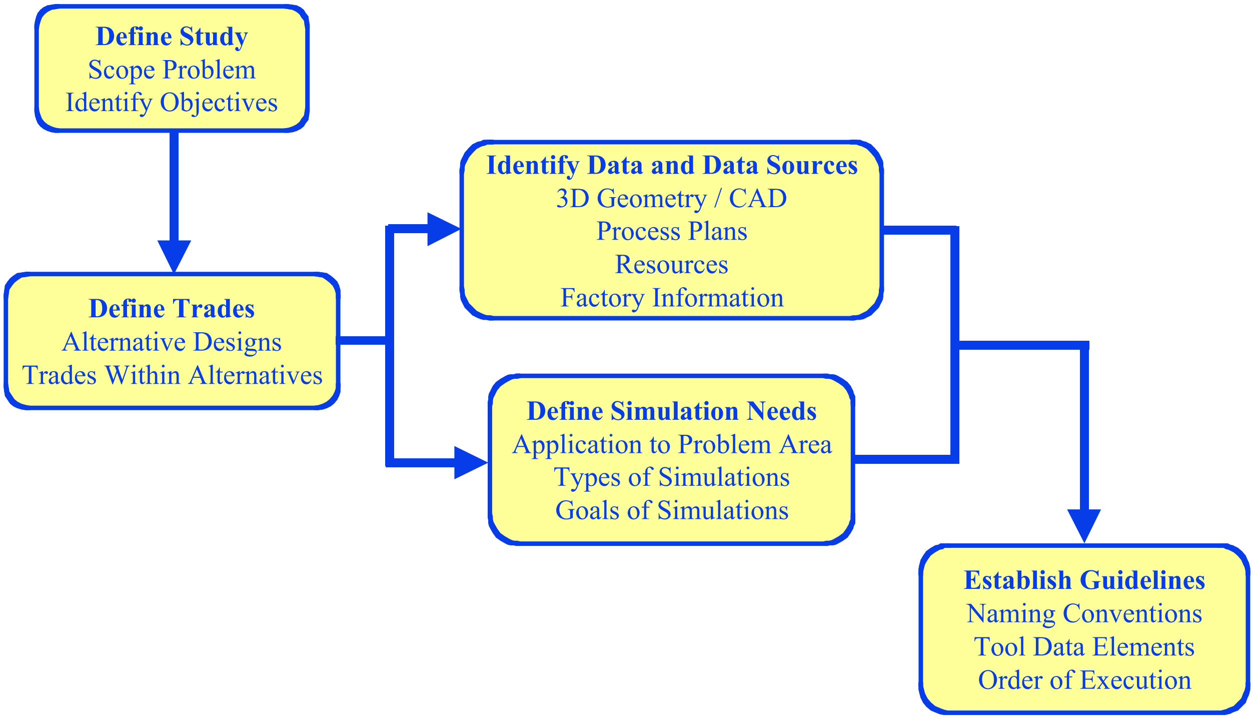


Figure 3-2: Up-Front Planning Activities

The SAVE usage scenario assumes that the development activity is being conducted utilizing Integrated Product and Process Teams (IPPT) which contain all of the disciplines involved in the design and manufacture of an item in a single, coordinated development team. The need for an assessment using SAVE is usually initiated with the identification of a project that requires a design effort. This effort may be a new design activity or a redesign resulting from some problem or issue. The IPPT that is assigned to the project needs to gain an understanding of the issues and to identify the specific objectives of the design activity. Once these objectives are well understood, the team will typically identify a series of trade studies whose results will lead to a preferred solution to the problem. At the highest level, these trade studies will identify a series of alternative design concepts that have potential benefits. As these concepts are developed, the team will likely identify numerous options within each alternative that require decisions prior to concept selection. For example, the team may want to evaluate the use of different materials in each concept. Other options might include fabrication processes or assembly sequences and techniques. In addition, the team will identify the assessments that are necessary to make comparisons among the various alternatives. SAVE does not require that more than one alternative be available for evaluation. The system may also be used to evaluate a single design at first, with options for improvements identified as a result of the simulations.

Once the alternative designs are identified, the IPPT will identify the data that is necessary to conduct the trade studies. The three-dimensional geometric definition is a key element for evaluating trade alternatives. Many simulation tools use this information whether in the form of actual geometry or as a list of features, parts and assemblies. Other key information for SAVE is the process planning data along with the associated resource and factory information.

In parallel with the data gathering activity, the team will make specific plans for the use of manufacturing simulation in the trade studies. First, the users must identify which types of simulations are appropriate for use in the defined problem area. For example, some design efforts may benefit more from assembly-level simulations than from part fabrication-level simulations. Figure 3-3 provides examples of the types of simulations and simulation results that are available from the tools within the SAVE environment.

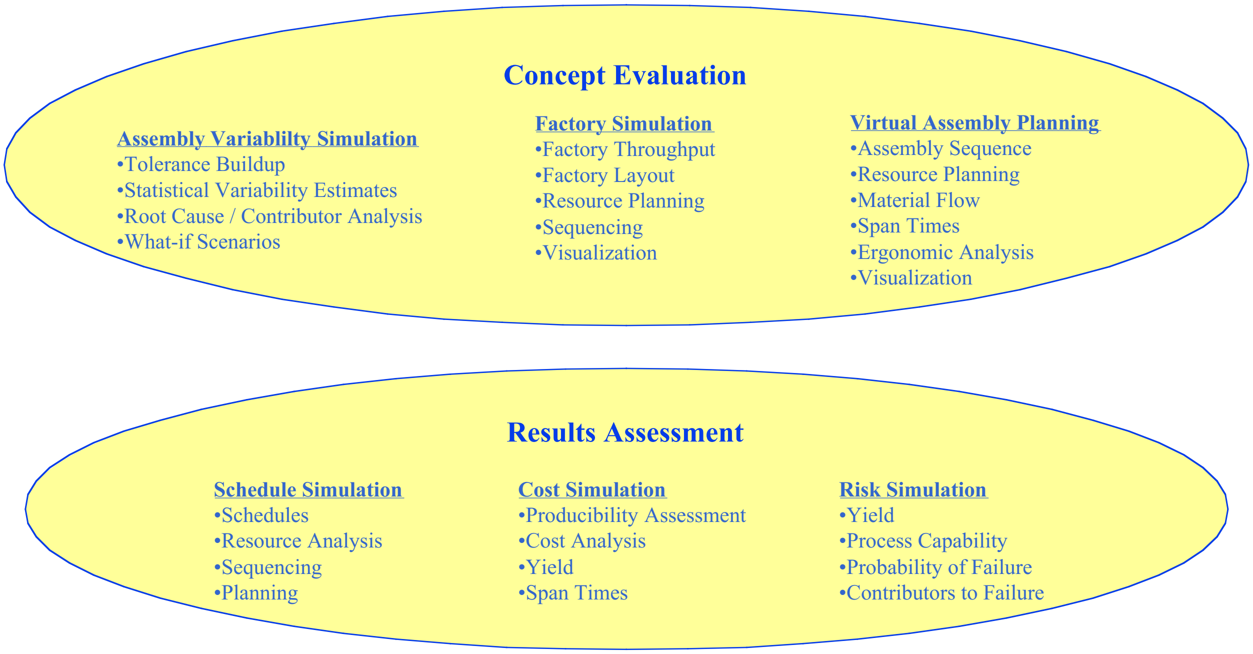


Figure 3-3: Examples of Simulations Types Available for SAVE

In conjunction with selecting which types of simulations to use in evaluating the design alternatives, the IPPT will identify the overall goals or types of information desired from the simulations. Most of the simulation tools within the SAVE environment can be used in a variety of ways. By identifying the goals of performing the simulations, the team will bring focus to the application of the tools. This focused planning will provide the basis for determining the required inputs and the desired outputs for each simulation.

After the team identifies the data requirements and defines the simulations, it will begin establishing guidelines for the use of SAVE in the design problem under consideration. This planning activity is critical to using SAVE effectively, because lack of agreement within the team to follow the guidelines will likely result in unnecessary rework downstream. Reaching agreement on these guidelines will require some understanding of the content and structure of the SAVE Data Model. The data model was developed with great flexibility so that it may be applied within any manufacturing domain regardless of product or company; however, with this level of flexibility comes a usage cost. The members of the IPPT will have three primary areas of coordination: naming conventions, tool data elements, and tool execution order. A brief summary of these coordination areas is provided in this section with more detail available in Section 3.3.

Prior to generating the initial process planning data for the design alternatives, the team should agree on naming conventions for everything from operations and resources to parts and CAD models. The SAVE data model itself does not restrict these names; however, individual simulation tools may have specific requirements. Users will need to identify those restrictions as well as any desires for cross-mapping among tools in order to develop an acceptable set of naming conventions.

Some of the simulation tools within the SAVE environment have overlapping capabilities. For example, Cost Advantage and IGRIP both have the capability to calculate times required for specific operations within a process plan. In fact, the overlap of data used and data generated by these tools is the primary justification for creating the SAVE integrated environment. The SAVE team has compiled a mapping matrix between the data elements of SAVE and the individual simulation tools. This matrix is available in Appendix O of this document. The IPPT will use this matrix to establish the responsibilities of each simulation tool. There may be situations where different tools may have responsibility at different stages of the trade study. Using the example above, Cost Advantage may make the initial estimates of operation times with updates from IGRIP later in the study. As these decisions are made, the IPPT must begin to identify the order of simulation tool execution.

There are several considerations in establishing guidelines for the order to perform specific simulations. The first, mentioned above, stems from the desire to have specific tools responsible for generating certain data elements. Another consideration is in making the most efficient use of the SAVE integrated environment. A significant benefit of integration, as opposed to simulation itself, is in the ability to quickly perform the simulations by sharing data among the tools. This notion of "create once, use many" is a key concept of SAVE. By carefully planning the order in which the simulation tools are used, the IPPT can reap the maximum benefit of integration. For example, using the simulation tool requiring the least amount of effort for data entry as the starting point for the series of simulations makes the most efficient use of the shared database.

3.2 Use of the SAVE Environment

Once planning activities are complete, the design team is ready to perform the design study using the SAVE environment. Figure 3-4 shows a typical activity flow for an IPPT conducting a design study.

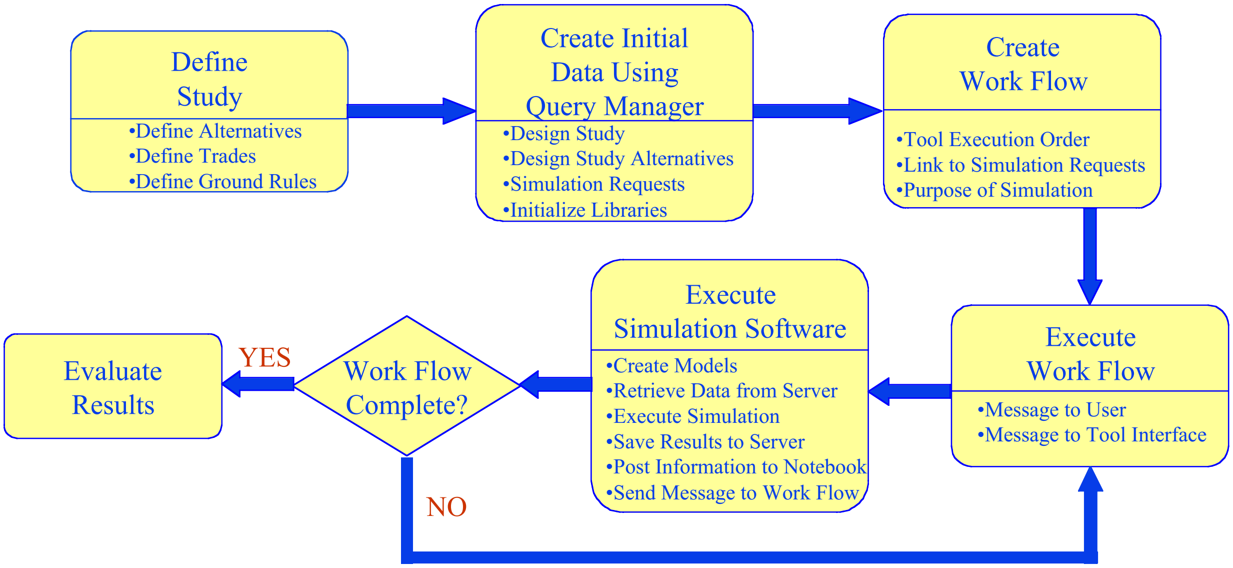


Figure 3-4: Typical Activity Flow for SAVE

The majority of the study definition tasks are conducted during the study planning phase described in Section 3.1. At this point the IPPT should have a clear outline of the design alternatives that will be considered, the trade studies that will be conducted, and the ground rules for a smooth study execution. With this information as a basis, the team is ready to effectively use the SAVE environment.

There are several objects in the SAVE data model that need to be populated by the IPPT prior to launching any of the simulation tools. These objects fall into two categories: model management objects and library objects. The SAVE toolsuite includes a Query Manager that allows the user direct access to the database for object browsing and creation.

The model management objects provide information to the tools about which information to access for their particular simulation activity. The IPPT creates design studies, design alternatives, process plan placeholders, and simulation requests. The design study defines the overall problem area that is being addressed. For example, a design study may be defined as the design of a horizontal stabilizer for a particular aircraft. In addressing the problem area, the IPPT has defined, as part of the planning activity, several design alternatives that identify the approaches to be evaluated and compared. For the horizontal stabilizer example, there may be a stiffener/skin option as well as an integrated structure option. Although the manufacturing simulation tools will typically populate the process plans for the design alternatives, the IPPT should create empty process plans within the design alternative as a placeholder for the information or copy an existing process plan that is similar to the one for this design alternative. If the IPPT identifies options within a design alternative, these may be developed as alternative process plans. If there are numerous options, the IPPT may choose to define a new design study and develop design alternatives within that study. The definition of these objects along with the fields that are necessary to define them is available in Section 2.0.

In order to provide information to the SAVE tools, the IPPT will create simulation request objects. These objects contain the information that is necessary for a SAVE tool to communicate with the SAVE database. It contains the design study, design alternative, and process plan that the tool will use. Several tools may share a simulation request if they are working with the same set of data. If the design team wants the tool to import from one process plan and to export its data into a separate copy of the process plan, the tool will need two separate simulation requests. This method may be useful if a simulation tool is making a lot of changes to the process plan. In this situation, the original process plan is left intact and an alternative process plan is created with the results of the simulation. The IPPT will have to make decisions about which process plan will be used by the downstream tools.

SAVE provides a number of library objects that may be used to collect and store information that is used frequently by the design team and the simulation tools. For example, material information may be common for a number of different parts. The library allows the material object to be populated once for a specific type of material and referenced by any number of parts that use that material. Ultimately, this library information will be obtained from existing databases within the companies implementing a SAVE environment. If the team identifies libraries that are not currently populated, the Query Manager may be used to build the individual library objects. Otherwise, the first tool to that needs the library will generate the object for use by subsequent tools. The following libraries are currently defined:

- Break
- CAD Model
- Design Alternative
- Design Study
- Inflation Table
- Material
- Manufacturing Program
- Part
- Personnel
- Process Plan
- Reference Process
- Simulation Request
- Tool
- Work Calendar
- Work Shift

Once the appropriate data is initialized using the Query Manager, the team is ready to build the initial work flow for the study. The Work Flow Manager provides a graphical interface for creating work flow models with three levels of decomposition. Although the Work Flow Manager has the capability to model the entire design study within a single work flow, the process will be more manageable if there is a separate work flow for each design alternative. In general, the IPPT will use the Work Flow Manager to define the process for each trade study. This includes defining the order in which the simulation tools will be used, providing information about the simulation request that contains the pointer into the data model, defining the results desired from the simulation, and linking the activities together in the desired manner. Guidance for using the Work Flow Manager is available in the Work Flow Manager User's Guide in Appendix L.

As the work flow is executed, the Work Flow Manager provides color-coded feedback to the design team that relates the status of a particular activity. These status messages are based on communication between the Work Flow Manager and the simulation tools. When a tool is notified to begin, there are two options for its execution. If the tool is capable of batch execution, it will import the specified process plan and automatically run the simulation. Once the simulation is complete, the appropriate data is exported to the SAVE database and a message is sent to the Work Flow Manager notifying it of the completed process. If the tool requires user interaction, an e-mail notification will be sent to the user. When the e-mail is sent, the Work Flow Manager will put the activity in a paused state until the user indicates his readiness to execute the simulation by manually resuming the appropriate activity through the WFM interface. At this point, the user will create any additional models, resume the work flow, and execute the simulation.

In addition to providing feedback to the Work Flow Manager, the user will likely want to communicate additional information to the entire design team. This is accomplished with the collaborative electronic design notebook. This collaborative communication environment allows the team member to provide a simulation results summary and any design recommendations resulting from the simulation. The notebook also provides a sort of running documentation of the design study.

Once the entire work flow is complete, the IPPT may use a combination of the Query Manager and the collaborative electronic design notebook to review the overall study results. At this point, the team may select a particular design or identify additional trade studies.

3.3 User Guidelines and Responsibilities

To achieve the maximum benefits from an integrated environment like SAVE, users must understand their tool's capabilities, limitations, and interface to the SAVE Data Model. Armed with this information a design team can plan an efficient order for tool execution, iteratively refining the details of the process plan, leading to accurate estimates of cost, schedule, and risk. Of particular importance is the initial population of the process plans and its operations. One tool must be chosen to perform the initial creation of the set of operations. Ideally, this tool should be capable of assisting the user in identifying the necessary operations and of populating a wide range of the supporting data, such as parts, materials, resources, etc. In some cases, it will take the execution of several tools to make initial estimates of the broad range of data. With initial estimates loaded, the process can be iterated to refine the information needed to support design decisions. For example, a detailed assembly simulation can be run to refine time estimates, which are then rerun in a schedule simulation tool prior to input into a detailed cost model.

Development of the full work flow should involve all team members, each of whom can identify their discipline's role and the quality of data that can be provided at each step. In many cases, there will be overlap in tool capability and the team must decide the most efficient path to take. As use of the integrated toolset grows, teams will have better understanding of the overall capability and these decisions will become easier. Work flow descriptions should be saved, as they can be reused or modified for similar problems.

3.3.1 Understanding the SAVE Data Model

SAVE users can be considered to fall into two categories, although in many cases one person will fill both roles. The first category includes the users who develop the simulation scripts or "models" (for example cost models) that are executed in the commercial simulation tools. Developing these scripts or models for tools that will communicate with the SDM requires that these users have a fairly detailed knowledge of the data objects supported in the SDM. End users, those who actually execute the simulations, must have some knowledge of the SDM, although to a lesser degree than the model generators. Some tools will have user-controlled input and output options and the users must select among these options for each tool execution to lead the design team to a complete, consistent set of decision data.

3.3.2 Naming Conventions

The flexibility that exists in the excellent commercial simulation tools has been reflected in the SAVE Data Model. No attempt has been made to constrain the names and descriptions of the key data objects such as the process plan operations, materials, or resources. While having an explicit set of these objects available might simplify integration in some cases, it is bound to be too restrictive in many more cases and is, therefore, unacceptable for a general integration solution. The final linkage of data among tools is dependent on simulation model developers, and to a lesser extent end-users, establishing naming conventions within their models that are ultimately written to the SAVE Data Model.

The required naming convention must, at a minimum, be established within a design study. It is more productive, however, to establish these standards across a complete design project, or the total design organization. The primary data objects that should be addressed include:

- Simulation Request
- Process Plan
- Operation
- Feature
- Material
- Reference Process
- Personnel
- Tool
- Part

The design team should define all attributes within the design study and design study alternative. These items help organize the trades that are being performed.

Simulation requests are defined by the design team in conjunction with setting up the work flow. This tells the tools what information to use in their simulation (design study, design study alternative, and process plan). It also defines the input/output options and any input files necessary to execute the simulation - if the options and files exist.

Design teams should standardize on the following information to make the simulation code interpretation of information consistent:

Process Plan: name, description

Operation: name, description, id number (team needs to decide how these items will be used/interpreted by the simulation tools)

The SAVE demonstration team used unique names for all operations and process plans because of requirements of some vendor tools. The uniqueness was accommodated by appending numbers to the "standard name" for the item. For example, there may be several drillream operations in a process plan. These were made unique by drillream01, drillream02, etc. Some tools that needed just the "category" of operation stripped the number and mapped the information appropriately.

Feature: name, type

Material: name, type, form

The simulation tools use this information in various ways and will typically look for specific names or types in mapping this data for their own internal usage.

If libraries are used, the reference process names need to match with the operation names that they refer to. Type could be used for this match, but the team needs to verify that it isn't being used for something else (like process versus simple operation).

Use of type in tool and skill in personnel needs to be agreed upon within all tools that use resources. Name attributes in these could also be used to differentiate.

Part naming and number conventions should be standardized and verified as valid inputs to the simulation tools. Some tools, for example, cannot accept a number as the first character of a string.

3.3.3 Overlap in Tool Capability

Among the tools used in the SAVE development effort, there are cases where more than one tool can estimate a given value in the SDM. Users must understand the capability of their chosen tools and control the refinement of data within the SDM by properly ordering tool execution and output options. In many cases one tool will be used to make a preliminary estimate of a value, for example an operation time, and later a different tool will update that estimate by using more refined inputs generated by other tools. Remember that many variables are stored as versioned variables and retain the complete history of estimates as they are refined during a design study.

3.3.4 Accessing the SDM Data

Most access to the data within the SDM will be accomplished though one of the commercial simulation tools. There are, however, parts of the model that must be manually populated. For example, design studies, design alternatives, and manufacturing programs must be manually created as most tools work at the process plan or lower levels. The SAVE system provides a Query Manager (QM) utility to allow users full capability to view, create, and modify data within the SDM. This query utility will ultimately be a web-based application, easily accessible through any web browser. The user interface is quite simple. Starting with a list of available SDM libraries, the user can easily traverse to or create any data object and operate on any attribute within that object.

The QM is used in conjunction with the SAVE Work Flow Manager (WFM) to set up design study work flows. As the work flow is graphically drawn in the WFM, the QM is used to create the simulation request data objects that are used to pass launch and data context information to the simulation tools.

3.3.5 Creation of Data Objects Within the SDM

If users plan to utilize the Query Manager to create objects, they should become familiar with the conventions used by the SAVE server in determining which attribute data is automatically created when an object is created. Most data objects contain other data objects as attributes, and users should understand that some, but not all, of these "sub-objects" are created automatically. Data Objects that are Named Objects (inherit from Named Object), that is have names and descriptions are NOT created automatically. These attributes must be manually created or found in libraries and used to populate appropriate attributes. Un-named, or base objects, are really just complex attributes (multiple fields or containing methods) and are automatically created and associated to the appropriate attribute. This convention is actually quite logical and will make sense, as the user becomes familiar with the SDM.

4.0 Metrics/Benefits Assessment

The potential savings from a well-implemented SAVE system are quite large. For example, estimates made at the initiation of the SAVE program projected a potential for a $1.1 Million per ship cost savings for the Joint Strike Fighter program.

Metrics measurement and validation has been an important element throughout the SAVE program. In fact, the significant opportunity for cost savings was the primary impetus for initiating the program. As a part of assessing the benefits of SAVE, the team identified the following seven key metrics which its Virtual Manufacturing technologies will impact.

- Design Change Reduction - Simulation allows verification of designs, planning, processes and tools prior to making actual pasts thus producing better designs with fewer errors that require redesign

- Scrap, Rework, Repair Reduction - Simulation allows verification of designs, planning, processes and tools prior to making actual parts thus producing better designs that have far fewer problems during production

- Design to Cost Accuracy - Accurate cost prediction methods allow better design choices to be made when performing trades between approaches which are close in cost.

- Lead Time Reduction - Process optimization leads to better schedules and closer to just-in-time inventory

- Process Capability - Assembly tolerance buildup is simulated and yield predictions are developed. This yield information allows assembly process improvements that reduce variation to be identified.

- Fabrication & Assembly Inspection Reduction - Simulation allows optimization of the inspection process, produces designs with fewer errors and provides better shop floor training aides.

- Inventory Turn Increase - Process simulation improves and reduces risk of implementing the "just-in-time" production concept where inventory is ordered and received as it needed instead of being stockpiled for future use. Systems can be pulled rather than pushed in production leading to reduced work in progress.

Users of the SAVE Virtual Manufacturing Environment will likely want to quantify the benefits the use of the system has on these and other measures of merit. Details for performing these calculations are included in the Business Case discussion in Chapter 4 of this document.

Chapter 4

Implementation Plan

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

CHAPTER 4 – TABLE OF CONTENTS

CHAPTER 4 – TABLE OF CONTENTS (Continued)

1.0 Scope

This chapter discusses issues that will be faced by organizations as they implement an integrated set of SAVE-compliant manufacturing simulation tools. Many of the SAVE implementation issues are common to all large-scale software systems, and these will be briefly discussed in Section 2.0; however, this write-up is not intended to be a detailed tutorial on these general topics. Emphasis is placed on the issues and tasks that may be unique to the SAVE system. Understanding these issues and the tasks that they imply will maximize the probability of a successful SAVE implementation.

The four major steps in SAVE implementation are discussed in Section 3.0 with respect to the viewpoints of three personnel categories. Section 4.0 of this chapter includes a notional schedule for the tasks described. This schedule is based upon assumptions about the scale of the implementation and is included only as a starting point for implementation planning. The assumptions used in the schedule are the same as those used and documented on the sample cost/benefit analysis discussed below.

The business case for SAVE implementation, including estimated costs and benefits, is discussed in Section 5.0. A sample spreadsheet used to estimate implementation costs and benefits has been created and results for an example implementation activity are included in Sections 6.0 and 7.0. The results shown are for a SAVE implementation on a project involving approximately 100 designers and a product with four major subassemblies. These cost estimates are representative of actual implementations, but should be reviewed and updated with refined inputs as the proposed use of SAVE is better defined. The software costs shown are only notional and can easily be replaced with accurate estimates through discussions with tool vendors. The benefits analysis is notional but realistic and is based on inputs that are consistent with the implementation costs.

A brief discussion of metrics measurement is included in Section 8.0. The implementation benefits are developed around metrics that have been historically measured to assess design and manufacturing process quality. However, these metrics tend to be statistical in nature and are difficult to assess from a pilot or limited implementation. Some approaches to this near-term validation problem are presented.

2.0 Challenges in Implementing SAVE

In today's environment, a SAVE implementation should be considered a medium scale software implementation problem. The fact that SAVE involves several tools and an integration environment makes it more complex than implementing a single tool, but it is certainly less complex than fielding a Product Data Management (PDM) or Enterprise Resource Planning (ERP) system. Within the scope of medium scale systems, the complexity of implementing SAVE will vary from site to site, dependent on:

- The extent to which simulation tools are already in use.

- The level of current tool / organizational integration.

- The range of tools to be integrated.
- The size of the design organization which will use SAVE.
- The extent to which SAVE will be integrated into the larger design environment.

Rapid progress in the capability of manufacturing simulation tools has occurred in recent years, but many organizations have not fully embraced their use. One reason for this limited use is minimal integration among the tools, a problem that SAVE directly addresses. But a design organization must still grasp the concept and potential of simulation before the benefits of integration will be appreciated. Organizations that have applied isolated tools and have personal experience with the inefficiency of repeated data reentry will certainly understand the benefits of integration more readily.

One of the biggest challenges faced in deciding to implement a system of SAVE-compliant tools is getting the multiple organizations usually involved in this range of tools to recognize that they can and should exchange their data in an iterative, real-time manner. In some large organizations, the tools currently integrated by SAVE are the responsibility of Design Engineering, Systems Engineering, Finance (Cost), Manufacturing Planning, and Tooling. Traditionally these organizations have developed systems that minimize their dependence on each other. Cost methods have been developed that are historically based and do not require timing estimates from planning or resource requirements from tool design. Factory scheduling does not utilize risk information that may be available for a particular unique design element. Design tolerance determinations are made in isolation of tool design and assembly process planning. Only through concurrent consideration of the interactions among all of these disciplines can a development team hope to identify better product and process designs and eliminate the costly errors now currently discovered and resolved in production.

The concept of Concurrent Engineering has helped teams to recognize the significant benefits of information sharing, but the tools to support this concept have been slow to develop. The availability of an efficient means of information sharing can open all organization to sharing their data and willingly accepting data from other organizations to aid their own calculations.

In general, the benefits of SAVE integration will increase as the number of tools that are integrated increases. SAVE has currently been tested with six classes of tools, but there is a high degree of confidence that the current information-sharing model will support any class of tool within the manufacturing simulation problem domain. The benefits of SAVE integration will still be apparent with a small number of tools if they are from different, competing vendors and do not have any inherent integration. For example, the Deneb assembly and factory simulation tools, Envision and Quest, are well integrated and little would be gained from SAVE integration of them alone. However, if an organization wished to use Deneb's assembly simulation and Tecnomatix's factory simulation tools, then SAVE integration would have a high payoff, and be much simpler than custom integration.

The size of a design organization will have some impact on SAVE implementation planning. Larger organizations will certainly present some additional challenges such as how to organize the simulation teams or how many Data Model Servers to utilize. The flexibility of the SAVE architecture is a double-edged sword. It can fit to many different requirements, but it will require some consideration to determine the best option for a given implementation. Details of these options are discussed below.

One of the major architectural features of SAVE is the ability for a simulation tool to access data from SAVE without regard to where the data are physically stored. This abstraction of data access allows data to be maintained in existing databases or PDMs without the data management issues created by replicating data in more than one system. In this way, it can be much easier to integrate SAVE into the larger design environment. An implementation site is not forced to use this feature. SAVE will store all data locally if desired, but in many implementations it will be desirable to have the SAVE server access its data from existing databases. This capability does not require reprogramming of the server, rather simply loading data to inform each data object or attribute about where the data are physically stored. Use of this capability implies an additional task in implementation, and this is more fully described in a section below.

3.0 SAVE Implementation Plan

The following sections of the implementation plan will be organized by the major phases in implementing SAVE.

1. Initial decision to implement

2. Pilot application

3. Planning for full-scale implementation

4. Implementing the full-scale system

Not all organizations will use all phases, but they are included for completeness. Within each phase there is a discussion of implementation from three perspectives:

- Management

- End users - these are the design team members who will operate the simulation tools

- Implementers - typically these are persons with IS experience

3.1 Initial Decision to Implement

3.1.1 Management Perspective

Most references on design systems implementation recognize that management commitment to the activity is the single most important element of a successful project. The multi-organizational nature of SAVE accentuates the importance of strong management support. Typically, one organization will identify the opportunity to improve their process using SAVE and will champion the cause to identify the set of tools that make sense for the overall design activity. Early involvement of management is vital to assure adequate participation by all organizations that can use or provide data to the design environment through SAVE. Not all organizations will opt to utilize the full suite of tools, but implementing a subset should be done through careful consideration as a lesser implementation reduces the potential benefits. The design functions that should be included in these early discussions include:

- Design
- Tolerance Analysis
- Costing
- Risk Analysis
- Process Planning
- Tool Design
- NC Programming

The SAVE Concept of Operations in Chapter 2 of this document describes the potential interactions among these functions. Management should require that the full range of possibilities be considered as part of the initial decision to implement.

A key decision that must be made in the early stages of planning involves whether to include a pilot implementation. A pilot should be considered if:

- The site has limited experience in the use of the simulation tools considered.
- Cost and benefit inputs need to be determined for a site-generated business case.
- There is a desire to validate SAVE within the larger design environment.

The choice and scope of any pilot should be carefully considered to assure that sufficient information is developed to support making the decision to fully implement the system. Quantification of the benefits by measurement of metrics is best accomplished by running the pilot in parallel with the traditional design process, but this can be cost prohibitive. In any case, it is best to select a design study that clearly has alternatives or issues involving cost and producibility. Demonstrations and pilots to date have been of approximately 3-6 month duration, involving 2-3 manyears of effort.

Many medium to large-scale design projects today involve inter-company teaming and virtually all projects include a number of major subcontractors or vendors who are involved in design of a portion of the product. A major decision that must be addressed by management is whether, and to what extent, to involve team members and vendors in the use of SAVE-integrated simulation tools. The SAVE system is inherently designed to operate in a networked environment and can be flexibly adapted to most situations. Small teams, with minimal security restrictions could use a single server, while larger teams or companies with secure firewalls may opt for one or more servers per site. These issues are discussed in somewhat more detail below and management should ascertain that the issues are resolved during this phase of implementation planning.

A notional schedule for the tasks involved in SAVE implementation is shown in Section 4.0 of this chapter. The implementation team can use this schedule as a starting point for developing their plans. All of the tasks discussed below are included and time estimates are based on a notional implementation within a design team of approximately 100 designers and an assumed project involving 4-5 major subassemblies.

A copy of an implementation cost estimation spreadsheet is also included in Section 6.0 of this chapter. The example shown is for a medium scale, full implementation. The actual spreadsheet is available in electronic versions of this document, or by contacting James Poindexter, Air Force SAVE Program Manager, james.poindexter@afrl.af.mil. With a minimal number of inputs, a quick estimate of the cost of a SAVE implementation may be computed. As more detailed information is obtained during planning, for example quotes from software vendors, default values can be replaced to improve the accuracy of the results. These implementation cost results are a major input into developing the business case for use of integrated manufacturing simulation.

3.1.2 End User Perspective

End users have the largest responsibility for planning an implementation leading to a decision to go forward. End users must recognize the advantages of integration and commit to changing their processes in order to share their data with other organizations and to use information from other groups to refine their own efforts. This is at the heart of the cultural changes that are often discussed as a major stumbling block to implementing revolutionary changes.

End users must become familiar with several things as they participate in the decision to implement SAVE. First, they must have, or develop, a thorough understanding of the capability of the simulation tools available to their discipline. Each discipline should also learn, at least at a high level, about the other disciplines being considered for integration. This information is best obtained in meeting with those disciplines, but is included in this report Chapter 2. The SAVE system supports information transfers among the various tools, but does not rigidly enforce a fixed set of transfers. To a large extent this flexibility was required by the flexibility and versatility of the simulation tools integrated by SAVE. End users must, therefore, understand the capability of the tools they wish to use as well as the contents of the SAVE Data Model, which implements the integration path.

This SAVE Data Model is documented for users in Appendix C of this report. This Data Model Dictionary is implemented as a set of hyperlinked web pages and is also available in that form on the SAVE website (http://skipper.mar.external.lmco.com/save). The web-based version can be accessed via either a list of the data objects or by a graphical index to the key objects. Each data object has a page describing that object and its associated data attributes and active methods if they exist. Users have found that the web pages are very analogous to the underlying object-oriented nature of SAVE. Their use simplifies the basic understanding of SAVE that is helpful in planning process plans that are at the heart of the SAVE model and of the design studies being performed.

The major tasks performed by the end users during this phase include:

- Lead the selection of the pilot design study (optional).

- Select the toolset, considering both commercial tools and any legacy tools that should be wrapped to work in the SAVE environment.

- Work with implementers to select appropriate computer platforms.

- Help develop estimates of hardware and software acquisition cost and training required.

- Work with implementers to develop a detailed schedule for the implementation. This schedule should include tasks for:

  - ✓ Pilot exercise if one is desired
  - ✓ Select tools and acquire software
  - ✓ Wrap legacy codes to be SAVE-compliant
  - ✓ Determine back-end data storage requirements
  - ✓ Train users on software - simulation tools and SAVE infrastructure
  - ✓ Select and acquire hardware platforms
  - ✓ Prepare a system test plan to validate installation
  - ✓ Install system and test
  - ✓ Gather data for design study - CAD models, existing manufacturing plans, etc.
  - ✓ Establish the naming conventions to be used project or company-wide to assure that all tools recognize common data elements. (This is discussed in detail in Chapter 3 of this report.)
  - ✓ Develop or modify required simulations and cost and risk models
  - ✓ Develop metrics plan
  - ✓ Execute design study
  - ✓ Measure metrics
  - ✓ Report to management

- Identify measurable metrics that can be used to track successful use of SAVE. (A discussion of metrics planning is included in Section 8 of this chapter.)

- Estimate the improvements that can be obtained in these metrics for inclusion in the business case.

- Develop a business case for SAVE implementation, including estimated costs and benefits.

The decision to include a pilot implementation of SAVE should be strongly considered by users to:

- Gain experience with the basic simulation tools.

- Investigate the efficiency provided by SAVE integration.

- Measure metrics chosen to validate the benefits portion of the business case / return on investment (ROI).

An early estimate of the cost of performing a pilot can be obtained using the spreadsheet discussed above and shown in Section 6.0 of this chapter.

3.1.3 Implementers Perspective

Most large organizations will have personnel skilled in planning software implementations. Where this is not true, one (or more) of the software vendors involved, particularly the vendor providing the infrastructure (Data Model Server and Work Flow Manager) can perform this function.

The major tasks performed by the implementation team during this phase include:

- Participate in the selection of the pilot design study.

- Work with the users to select the toolset.

- Work with vendors to select appropriate computer platforms.

- Become familiar with the concepts of the Common Object Request Broker Architecture (CORBA) in general, and the particular vendor's ORB. This may involve training.

- With inputs from users, develop estimates of hardware and software acquisition cost and training required.

- Estimate continuing computer support required.

- Work with users to develop a detailed schedule for the implementation. (See details in User Perspective.)

3.2 Pilot Exercise Planning and Execution

3.2.1 Management Perspective

The decision on whether or not to perform a pilot application of SAVE should have been made as part of reaching the decision to implement. Including a limited scale pilot could be included as a "Go/No-Go" decision point for full-scale implementation, or simply as a prudent learning exercise before implementing project or company-wide. In either case, the planning used to reach this decision provides the basis for the pilot study. Management responsibility during this phase is common to most company activities:

- Authorize adequate personnel and budgetary resources.
- Establish a periodic status reporting mechanism.
- Assure that metrics are measured and reported.

Particular attention should be paid to any indications of cultural barriers impeding full acceptance of the integration that is the objective.

3.2.2 End User Perspective

A pilot design study will typically involve a single design team and a well-bounded problem. The design team will include some or all of the following:

- Designer
- Cost Analyst
- Risk Analyst
- Tolerance Analyst
- Process Planner
- Tool Designer
- Manufacturing Engineer

Depending on the size of the problem chosen, the tools selected for the pilot and individual experience, some team members may perform multiple roles, and not all members may be full time.

If required, team members should receive training for the tools they will be responsible for. Training in the use of the SAVE system (Work Flow Manager, Data Model Interactive Access, and Electronic Design Notebook) are all relatively simple and should be obtained from one of the SAVE-compliant tool vendors or implementation consultants.

The design exercise should be planned as a team. As part of this process, team members will determine the execution order of the simulation tools in order to create and refine the alternative process plans and their associated cost, schedule, and risk estimates. In determining the execution order of tools it is helpful to think in terms of populating and refining the process plan, which is the key data structure within the SAVE Data Model. During a pilot study, the end users will accomplish the following steps:

- Identify design study with alternatives.
- Determine order of execution of simulation tools.
- Capture plan for study in Work Flow Manager.
- Gather data for design study - CAD models, existing manufacturing plans, etc.
- Establish the naming conventions to be used project or company-wide to assure that all tools recognize common data elements. (This is discussed in detail in Chapter 3 section of this report).
- Develop or modify required simulations and cost and risk models.
- Execute design study.
- Record team information (schedules, results, decisions, etc) in electronic design notebook.
- Measure metrics.
- Report to management.

3.2.3 Implementers Perspective

Understanding the issues of the computer implementation tasks will be a key part of any pilot exercise. The implementation team is responsible both for bringing the pilot to operational status, but also to be prepared to do detailed planning for full-scale implementation. SAVE-compliant tool vendors will provide a valuable resource to the implementation team.

Most of the implementers tasks during a pilot exercise are common to any software system implementation. Many pilots will focus on the basic functionality of SAVE, but some pilots may also be planned to develop an understanding of writing SAVE-compliant wrappers for company-developed tools or to test tailoring the Data Model Server to access data from existing databases or PDM systems. Both of these topics are discussed in detail in other sections of this and the SAVE Software End Item reports.

The implementation team must accomplish the following tasks during a pilot exercise:

- Acquire software.
- Arrange user training - simulation tools and SAVE infrastructure.

- Select and acquire hardware platforms.
- Install system and test.
- Optional - Develop a SAVE-compliant wrapper for an internally developed tool or tools.
- Optional - Adapt the Data Model Server to store some data in an existing database.
- Report to management.

3.3 Planning for Full-Scale Implementation

3.3.1 Management Perspective

Planning for full-scale implementation is really similar to the tasks discussed above under Initial Decision to Implement, and differs only in the scale of the problem and the level of detail of the planning. When a pilot exercise is performed this detailed planning should be accomplished in parallel with the pilot in order to prepare for the decision to commit to full implementation.

Detailed planning will ultimately allow the implementation to progress smoothly and result in a minimum cost system. Management should provide adequate schedule and budget for this activity. There is no simple answer to the schedule and resource requirements for this plan. The range of implementations can span from single design teams to project-wide and even worldwide installations. A sample plan with estimated resources is included later in this chapter. In larger implementations more calendar time must be allowed to obtain buy-in from the wide range of using organizations and the plan should allow for a phase-in of the system.

Other than simple scaling, the major difference in a large implementation will be decisions regarding the number and location of Data Model Servers. This issue is discussed in the Implementers perspective section below.

3.3.2 End User Perspective

Planning for large-scale implementation will require users to make a more detailed survey of the tools that will be included, particularly if a wide range of sites will be involved. In many cases organizations will have adopted different tools for the same task at different sites and consideration of standardization must be made. The open, industry-standard approach to integration taken by SAVE would allow a wider range of tools to be maintained at lower integration maintenance costs, but standardization still has many advantages and should be considered. Users will also have to identify internally developed tools which will have to be wrapped to be SAVE-compliant. Plans must include resources for implementers to develop wrappers for these tools. In some cases, replacement of these tools with commercial off the shelf (COTS) solutions can be considered. Adequate time should be allowed to review and make these decisions.

The SAVE Data Model server is designed to allow data integrated by SAVE to be stored in the local SAVE-controlled data store or accessed through SAVE but stored an existing PDM or database. This feature avoids data replication and its associated data management issues. End users will be strongly involved with the software implementers to identify which type of "back end" to use. SAVE allows these decisions to be made with fine granularity - individual data fields of data objects can be stored separately, if needed. In many cases, this data storage relocation is defined by data stored in the SAVE Data Model server, not by a process of re-coding the server. Commercial SAVE servers may differ in this area, and the implementation team must discuss the capability with any potential vendor.

3.3.3 Implementers Perspective

Most large-scale implementation tasks are similar to the pilot tasks described above except for the scale of the effort. There are, however, two additional issues that must be addressed during planning for larger scale sites. The first issue is to decide on the number and location of Data Model servers that will be implemented. The SAVE architecture is strongly network-based allowing servers to be placed virtually anywhere in the system. Selection of the number of servers and their location are made based on performance and maintenance issues.

SAVE compliance in a server is defined by the CORBA-based interface to the Data Model. Server implementations can vary widely and the implementation team will need to discuss their requirements with vendors to ascertain the best server for their site. A single server is preferred if data storage scaling and network speeds allow. Multiple servers may be required in large or multi-site installations. Centralization of multiple servers is beneficial from a maintenance perspective but may not provide the performance users require. The majority of data in a SAVE server will be unique to the design team that generated it and only the reference information such as Reference Processes, Materials, Parts, and standard Resources need to be shared among teams and their servers in a large installation. The vendor who provides the SAVE server should be able to provide all of the information required to make the necessary decisions.

The implementation team will be heavily involved in the decision and implementation of the SAVE capability to perform physical data storage in a number of back-end systems including PDMs or existing databases. This is another area in which commercial SAVE servers can differ in their implementations and the implementation team will be responsible for studying the available servers and recommending the best solution for the site's overall architecture. The preferred approach for this data redirection is for the data element's storage location to be defined by data elements stored in the SAVE server, rather than requiring the server to be re-programmed and recompiled. This approach was designed into the original SAVE contract server and should be available in any commercial server.

Working with the end users, the implementation team will decide which of the data defined in the SAVE Data Model will be stored locally and which will be redirected to existing storage systems. These systems may include existing Product Data Management (PDM) systems or existing relational (or other) databases. A thorough understanding of the SAVE Data Model is required to accomplish this task. This SAVE Data Model is documented in Chapter 3 and Appendix C of this report. This Data Model Dictionary is implemented as a set of hyperlinked web pages and is available in that form on the SAVE website. The web-based version can be accessed via either a list of the data objects or by a graphical index to the key objects. Each data object has a page describing that object and its associated data attributes and active methods if they exist. Users have found that the web pages are very analogous to the underlying object-oriented nature of SAVE.

Once these data storage decisions have been made, the implementation team will make the formal specifications of the storage locations to the server in the manner prescribed by the particular server they have chosen. A test of this portion of the installation should be planned prior to releasing the system to the end users for production.

3.4 Implement Full-Scale System

3.4.1 Management Perspective

With the full-scale implementation plan developed in the last phase, management's focus during the actual implementation is to track that schedules and budgets are adhered to. Once the system is fully operational, management should assure that the decided upon metrics are measured. These metrics will be important both to validate that the systems is achieving the desired results, and to build the new historical database for cost projections of design and production resources.

3.4.2 End User Perspective

For end users, full-scale implementation involves initial training with the core simulation tools and the SAVE infrastructure, loading reference information into the SAVE Data Model reference libraries, and initial application of the system to actual design studies.

Some of the activities that began during planning for full implementation may continue into the initial use of the system. Development and refinement of the cost and risk models and creation of the actual manufacturing simulations will be continuing tasks. If a continuing metrics measurement and validation plan was adopted, the process to gather and report these data will be created and followed. Metrics should be documented in a fashion that makes them readily available for use in forecasting schedule and resources for future design and manufacturing programs as well as validating the success of the current installation.

The SAVE Data Model contains a small number of libraries that contain reference information not specific to a particular design study. Examples of these reference data include Materials, Reference Processes, Tool descriptions, Personnel resource definitions, and Work Calendars. The libraries are fully described in Appendix C of this report. These data can be gathered from existing sources and populated in SAVE on an as-needed basis, may simply be accessed from an existing database, or can be loaded as a one-time operation. Users will need to consider these options and perform the appropriate tasks as part of implementation.

3.4.3 Implementers Perspective

The full-scale implementation team obviously plays a major role in this phase of implementation by performing the following tasks:

- Acquire hardware.
- Acquire software.

- Install and test the system.

- Populate the server with information defining back-end data storage locations, if this option is chosen.

- Wrap any legacy tools that have been identified to work in the SAVE environment.

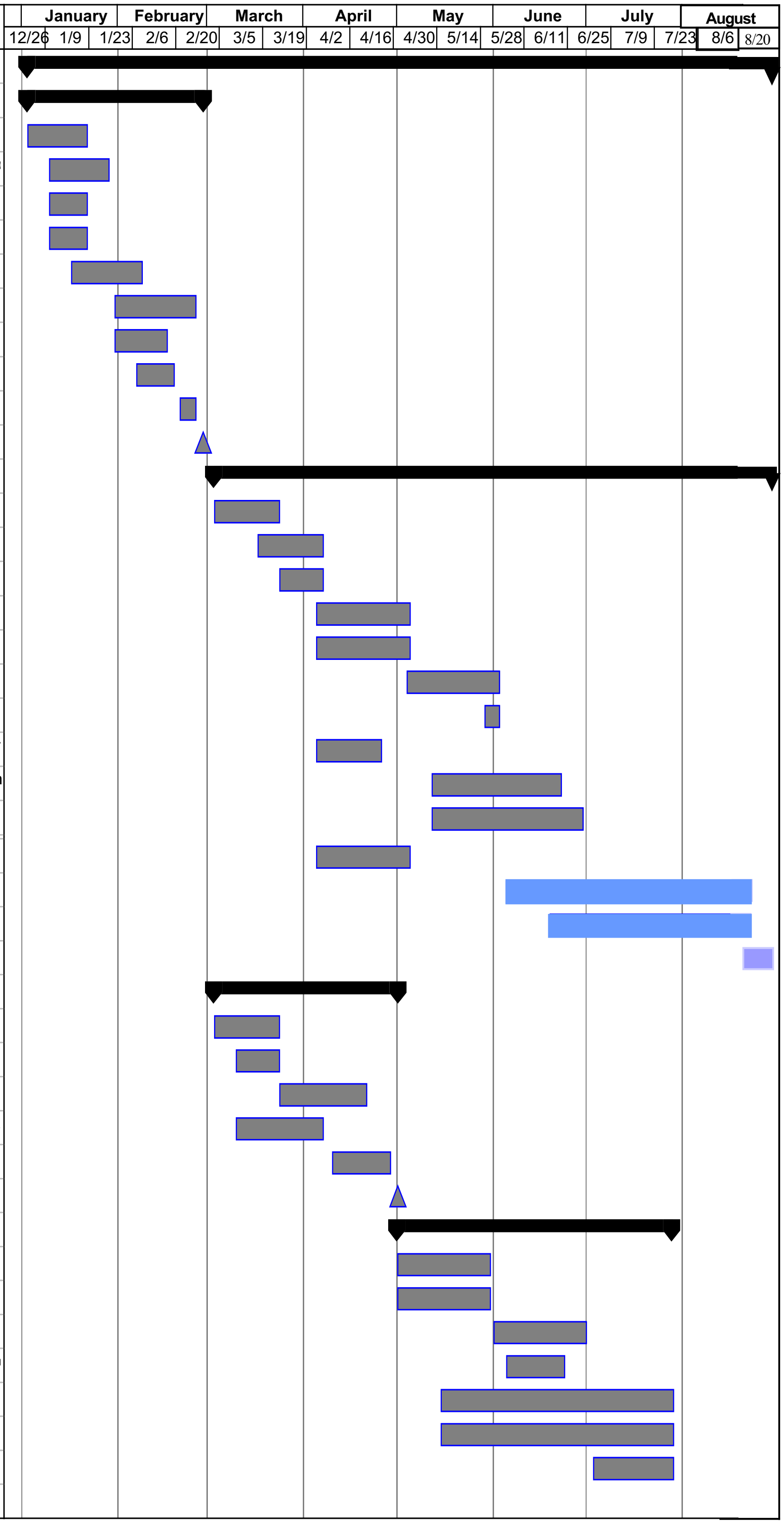
The first three tasks are standard for any software implementation and involve standard practices. The last two tasks are peculiar to SAVE and are discussed in more detail below.

The SAVE architecture was specifically developed to allow data that are accessed from the SAVE Data Model to be physically stored in any of a number of back end systems. There is a more complete discussion of this capability in the SAVE Software End Item Report. Competing commercial SAVE servers are free to implement this capability differently, although they all present the same CORBA interface to client codes. It is expected that mapping data object attributes to physical storage will be accomplished by adding data to the server rather than by recoding and recompiling the server. This task then involves identifying the desired back end storage systems and mapping the desired attributes to those databases. Most commercial vendors of server software will provide consulting services to aid this process.

During the planning process the end users may have identified legacy noncommercial tools that are to be integrated by SAVE. If this is the case, the implementers will have the task of wrapping these tools to provide SAVE-compliant, CORBA-based access to the server and to provide the required work flow manager server functionality. The wrapping function is discussed in full detail in the SAVE Software End Item Report. The task of wrapping a tool will vary with the complexity of the tool and the amount of SAVE Data Model data that the tool will read and write. Wrappers developed to date have required between 300 and 600 man-hours to develop.

4.0 Notional Schedule

ID	Task Name
1	**SAVE Implementation Tasks**
2	**Reaching Decision To Implement**
3	Determine design teams that will use SAV
4	Determine approach to team members an
5	Select classes of simulation tools
6	Select hardware platforms
7	Determine wrapping requirements for lega
8	Produce implementation plan
9	Consider pilot exercise
10	Produce business case - ROI
11	Prepare for and brief maagement
12	Decision to implement
13	**Perform Pilot Exercise**
14	Indentify design problem with alternatives
15	Select Tools
16	Select hardware platforms
17	Acquire software
18	Acquire hardware
19	Install system and test
20	Train users
21	Establish naming conventions for pilot tear
22	Develop or modify simulations and cost an
23	Gather data for design study - CAD model
24	Develop metrics measurement plans
25	Execute design study
26	Measure metrics
27	Report to management
28	**Plan Full Scale Implementation**
29	Refine tool selection
30	Validate hardware selection
31	Establish naming conventions for project /
32	Determine requirements for back-end data
33	Plan phased release to design teams
34	Report to management
35	**Implement Full Scale System**
36	Acquire software
37	Acquire hardware
38	Install system and test
39	Populate server with back-end data locatio
40	Develop or modify simulaitons and cost an
41	Wrap legacy tools if required
42	Train users
43	



5.0 Creating the Business Case for SAVE

Developing a solid business case for SAVE will be an important part of implementation planning at most sites. With so many technological advances occurring so rapidly, it is easy to become overwhelmed, making decisions on which technology and when to implement difficult. What may appear clear-cut to developers and some end users must still be sold to other end users and management.

A convincing business case will have to be tailored to each site considering SAVE implementation. As discussed in this section, the elements of both the cost and benefits are dependent on the specific status and capabilities of a development / production organization. Implementation costs will vary with the extent to which manufacturing simulation is already in use. Benefits will also be a function of how much simulation is in use and the historical design error rates, among other things.

5.1 Implementation Costs

Implementation costs are a function of many variables, and inputs are required from both end users and implementation personnel. A sample spreadsheet that can be used to estimate SAVE implementation cost is included in Section 6.0 of this chapter. Most inputs are self explanatory and are summarized below:

1. End User Inputs
   - Number of designers on design team
   - Number of manufacturing engineers (ME) on design team
   - Number of major parts in assembly
   - Number of major subassemblies
   - Man-hour wrap rate
   - Number of legacy tools to wrap
   - Include a pilot exercise?
2. Implementers Inputs
   - Training man-hours per tool
   - Number of backend data stores
   - Number of data objects remotely stored
   - Number of servers
   - Cost of server H/W platform

- Installation man-hours per simulation tool
- Installation man-hours per server
- Average cost of PC for simulation tool
- Average cost of UNIX platform for simulation tool

3. Costs obtained from S/W vendors

   - Server
   - Work Flow Manager
   - Query Manager
   - Cost Tool
   - Risk Tool
   - Assembly Simulation
   - Factory Simulation
   - Computerized Process Planner
   - Tolerance Analysis
   - Electronic Design Notebook, per user

4. Other Assumptions

   - Fraction of MEs performing simulations
   - Average size of simulation team
   - Estimated hours to wrap one legacy code
   - SAVE infrastructure training hours per user
   - Cost to implement remote storage for 1 object

This spreadsheet was developed to require a moderate number of inputs that can be easily gathered during the Initial Decision to Implement Phase to aid that decision. Reasonable estimates for all inputs are included with the spreadsheet. It should be considered a good starting point, but can be extended to more detail if desired. Section 6.0 shows the inputs and results for a sample estimate based on a medium size design team that involves approximately 100 designers, 60 Manufacturing Engineers and a product with 1000 parts in 4 major subassemblies.

The Microsoft Excel spreadsheet can be obtained by contacting James Poindexter, Air Force SAVE Program Manager (james.poindexter@afrl.af.mil).

Note that the spreadsheet produces the costs broken into two categories:

- Cost of implementing simulation tools
- Cost of implementing SAVE-compliant integration

This was done to address a specific request of the SAVE Advisory Boards at the June 1999 meeting to separate the costs and benefits of the simulation tools themselves versus the SAVE integration. The benefits discussion and spreadsheet also address these categories to aid in the two-level implementation decision – simulation and/or integration.

5.2 Integrated Manufacturing Simulation Benefits

The other side of the business plan involves the benefits of SAVE. Their estimation is somewhat more problematic. The approach to this assessment follows the metrics that were identified early in the SAVE development effort. Each of these metrics is briefly described below with a discussion of the SAVE benefits estimation spreadsheet immediately following.

5.2.1 SAVE Metrics

The following areas were identified as being the key metrics that would be improved by implementing a suite of integrated manufacturing simulation tools.

- Design Change Reduction

  This metric measures the reduction in redesign which results from errors and inadequate consideration of producibility and manufacturing costs. Estimates of the benefits in this area are calculated by knowing the historical quantity of design changes per part per year and the average cost of a design change. In estimating the impact of manufacturing simulation it is important to account for the benefits derived from other technologies such as 3-D CAD and digital mockup. Measuring a reduction in design error relative to historical levels validates improvements in this metric.

- Design to Cost Accuracy

  The objective of this metric is to produce consistent, accurate cost estimates of close, competing product and process alternatives. Ability to reliably choose between alternatives directly relates to cost estimation accuracy. Manufacturing simulation can have a strong impact on costing accuracy if time estimates, risk assessments, and resource requirements are included in cost estimating relationships, rather than simply using historical or weight-based methods. Comparing estimated cost to cost measured on the production floor is the way to validate improvements in this metric.

- Scrap, Rework, Repair Reduction

  This metric is aimed at measuring reductions in scrap, rework, and repair (SRR) which result from errors and inadequate consideration of producibility and manufacturing cost. The savings can be estimated knowing the historical percentage of SRR based on unit product cost and an estimate of the impact of integrated manufacturing simulation tools. Similar to the Design Change metric, it is important to account for the benefits derived from other technologies such as 3-D CAD and digital mockup. An organization that currently tracks SRR and categorized causes will find it easy to assess potential improvements from each of these design technologies. Measuring SRR after implementing SAVE will validate this metric.

- Fabrication & Assembly Inspection Reduction

  This metric quantifies the benefits of reduced fabrication and assembly inspection that results from developing simpler, higher quality manufacturing tools and processes. This metric can be quantified by knowing the historical cost for inspection as a percentage of production cost and applying an improvement factor estimated for SAVE. The factor currently used was estimated by members of the F-22 Advanced Tactical Fighter Integrated Product Development Teams. Tracking future inspection requirements against historical levels is used to validate improvements in this metric.

- Inventory Turn Increase

  This metric addresses the savings that can be achieved by reducing inventory cost by eliminating non-value-added activities and reducing fabrication and assembly process times. Measuring this metric involves estimating the financial cost of carrying the portion of inventory that is not actively being processed. Many companies currently track this metric, and validation of improvements due to improved manufacturing processing can be clearly measured.

In the development of these metrics, the SAVE system and its capabilities were described to members of the F-22 Advanced Tactical Fighter program design Integrated Product Teams and they (not the SAVE developers) estimated the factors used in the equations used to estimate improvements in the metrics.

5.2.2 Benefits Estimation Spreadsheet

The benefits calculations are based to a large extent on the early metrics estimations made during the early stages of SAVE development. The equations developed for the early metrics study provide the starting basis for the benefits spreadsheet presented here. The equations have been improved in some cases and the input values have been generalized where it was felt that they were specific to the F-22 design.

The primary inputs to the metrics have been consolidated and now form the input fields of the benefits estimation spreadsheet. These inputs include:

- Number of SAVE system users
- Development Program Span - Years
- Estimated Product Production Cost
- Number of Parts
- Number of Major Assemblies
- Estimated Material Cost
- Estimated Fabrication Cost
- Estimated Assembly Cost
- Production rate per year
- Number of units to count for savings
- Number of years to count for yearly savings
- Simulation Team efficiency increase due to SAVE
- Historical Error Rates/Cost
- Number of changes per part per year
- Average cost per design change
- Historical percent scrap, rework, repair

A few additional inputs are required in some of the benefits sections. Sample benefits results are shown in Section 7.0 of this chapter. The case shown corresponds with many of the assumptions made in the inputs to the implementation cost estimate. The inputs shown are believed to be reasonable estimates. A small number of cases of differing sizes have been run and all results appear realistic. Again, remember that this spreadsheet is provided as a starting point, and should be reviewed and changed or expanded if desired.

Implementation benefits are provided in two categories, for the basic simulation tools and for the SAVE-developed integration, as was done for the implementation costs. This distinction is made to respond to the request for separation of the benefits made by the SAVE Technical Advisory Boards. The benefits of SAVE integration can simply be viewed as savings in the man-hours needed to perform a series of manufacturing simulations. There are two ways to take advantage of this increased efficiency. The first way is to simply take a man-hour cost reduction. The second way can be much more powerful in most cases. In this second approach, the efficiency is used to allow more simulations to be performed with a fixed level of manpower. In cases where additional design studies can be identified, this second approach will generally be the best approach. The benefits spreadsheet calculates the potential benefit for both tactics.

Particular attention should be paid to the cost model accuracy metric. Improvements in cost model accuracy are shown to have a powerful influence on production cost savings by providing accurate decisions between design alternatives. SAVE integration is crucial to the success of detailed cost models. Integration makes accessing the needed part feature data, simulation times, resources utilization, risk data, etc practical. Without SAVE integration, the excessive manual data input effort would preclude the use of the more accurate cost models.

A simple statistical analysis was used to estimate the percentage of correct cost-based decisions between two alternatives, when cost methods of different levels of accuracy are used. A table of values was generated based on this analysis and is used, along with user inputs of average differences between alternatives and historical cost model accuracy. The full statistical analysis spreadsheet is included with the cost/benefits spreadsheet, although only the results are uses in this report.

6.0 Implementation Cost Estimation Spreadsheet

SAVE Sample Implementation Cost Estimate

Large Design Team Example

Primmary Inputs

🟨	Primary input fields
🟦	Secondary input fields - Supplied values are felt to be adequate
🟧	Calculated, but may be overwritten

Management Inputs		
Estimate Adjustment Factor (EAF)	1	Historical factor used to adjust technolgist's estimates to reality.

End User Inputs		
Number of Designers on Design Team	100	Provides basis for sizing based on headcount
Number of MEs on Design Team	60	Calculated as 60% of the number of designers
Number of Major Parts in assembly	1000	Not used - Can provide basis for sizing based on complexity
Number of Major Subassemblies	4	Provides basis for sizing based on design complexity
Manhour wrap rate	100	Average cost of a manhour
Number of legacy tools to wrap	2	Number of company proprietary tools made SAVE-compliant
Inlcude a pilot exercise ?	Y	This flag controls whether pilot exercise costs are included

Implementers Inputs		
Training manhours per tool	40	Average training manhours for one software tool
Number of backend data stores	3	No. of storge locations for SAVE data in addition to server
Number of data objects remotely stored	10	Examples: Materials, Reference Processes, BOM, etc
Number of Servers	1	Number of SAVE servers to be implemented
Cost of Server H/W Platform	20000	Estimated cost of a SAVE server H/W platform
Installation Manhours per sim tool	8	Estimated manhours to install simulation tool
Installation Manhours per server	40	Estimated manhours to install and test SAVE server
Average cost of PC for simulation tool	2500	Cost of typical PC platform for simulation tool
Average cost of UNIX platform for sim tool	12000	Cost of typical UNIX platform for simulation tool

Costs obtained from S/W vendors	**Platform**	**S/W Cost**	**Training Hours**	**Off-site train Per person**	**On-site train Class of 10**	**Break-even class size**
Server	Server	20000	N/A (Included in Infrastructure training)			
Workflow Manager	Browser	4000	N/A (Included in Infrastructure training)			
Query Manager	Browser	5000	N/A (Included in Infrastructure training)			
Cost Tool	PC	12000	40	2000	15000	5
Risk Tool	PC	1000	40	2000	15000	5
Assembly Simulation	UNIX	50000	40	2000	15000	5
Factory Simulation	UNIX	15000	40	2000	15000	5
Computerized Process Planner	PC	20000	40	2000	15000	5
Tolerance Analysis	UNIX	25000	40	2000	15000	5
Electronic Design Notebook, per user	Browser	100	N/A (Included in Infrastructure training)			

Note: PC/UNIX under Platform controls hardware cost est below

Other Assumptions		
Fraction of MEs performing simulations	0.5	Used to calculate SAVE users based on number of MEs
Average size of simulation team	6	Used to calculate SAVE users based on design complexity
Estimated hours to wrap one legacy code	500	Manhours
SAVE Infrastructure training hours per user	8	Manhours
Cost to implement remote storage for 1 object	160	Manhours

Calculated Values		
SAVE users based on # of MEs	30	SAVE users on design team based on headcount
SAVE users based on # of assemblies	30	SAVE users on design team based on design complexity
SAVE users based on average of above	30	Average of these two estimates is used for sizing below

Summary Cost Results

Total cost for basic simulation tools	**$1,889,044**	
Total cost for SAVE integration	**$465,116**	
Total cost	**$2,354,160**	
Yearly cost of MEs involved with Simulation tools	**$5,760,000**	
SAVE cost as percentage of one year	**8.1%**	Approx efficiency increase needed from SAVE integration to reach break-even in one year based on efficieny increase only

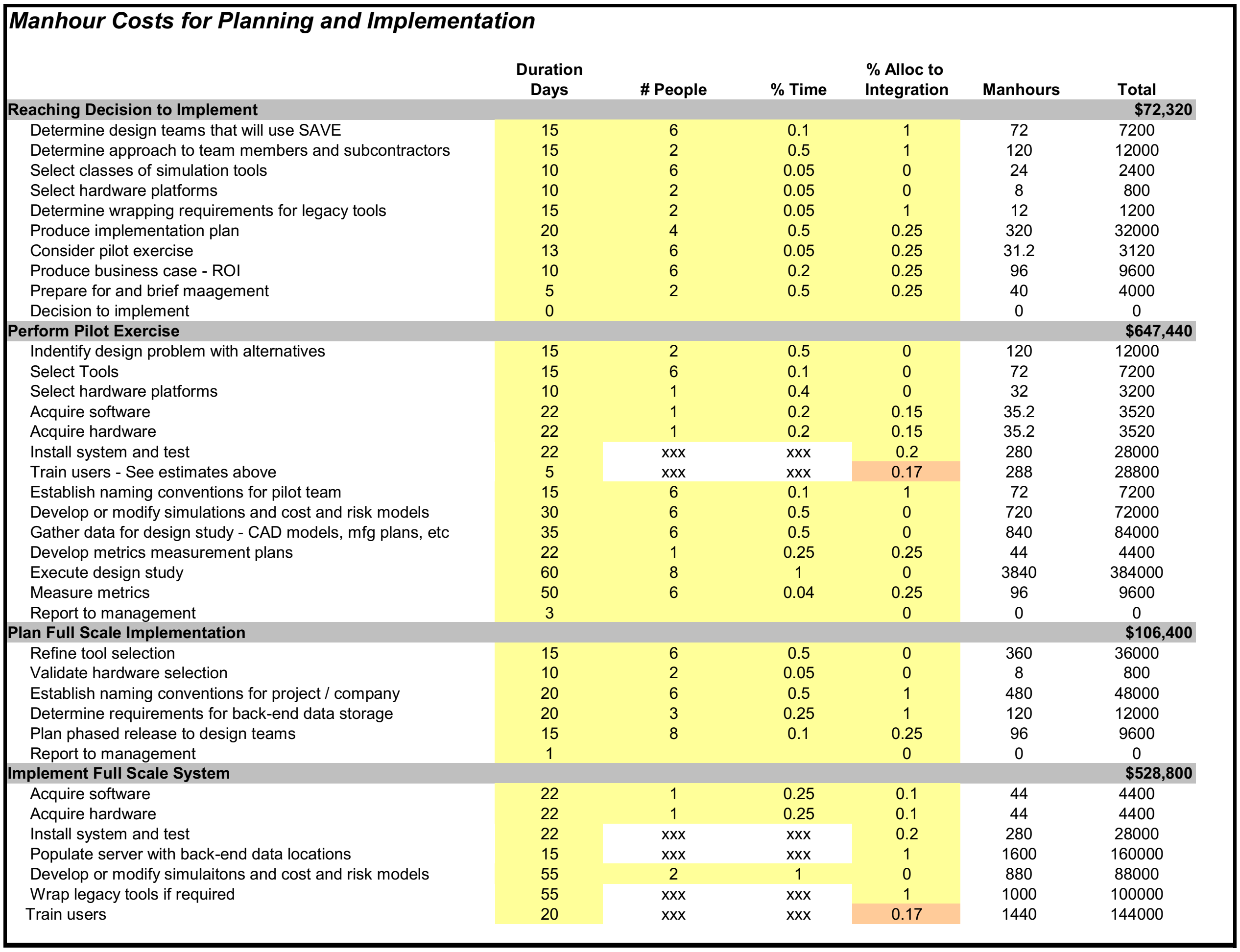
Non Labor Expenses

	I - Integration S - Simulation Category		Per Copy	# Users	# Copies	Total
Commercial Software						**$647,000**
Server	I		20000	30	1	20000
Workflow Manager	I		4000	30	1	4000
Query Manager	I		5000	30	1	5000
Cost Tool	S		12000	5	5	60000
Risk Tool	S		1000	5	5	5000
Assembly Simulation	S		50000	5	5	250000
Factory Simulation	S		15000	5	5	75000
Computerized Process Planner	S		20000	5	5	100000
Tolerance Analysis	S		25000	5	5	125000
Electronic Design Notebook	I		100	30	30	3000

		Platform	Per System	# Systems		Total
Hardware						**$237,500**
Sever Platform	I	Server	20000	1		20000
Cost Tool	S	PC	2500	5		12500
Risk Tool	S	PC	2500	5		12500
Assembly Sijmulation	S	UNIX	12000	5		60000
Factory Simulation	S	UNIX	12000	5		60000
Computerized Process Planner	S	PC	2500	5		12500
Tolerance Analysis	S	UNIX	12000	5		60000

	Category	# Heads	Manhours	Travel Exp	Course Cost	Total (Excludes MHrs)
Pilot Training Expenses						**$21,700**
SAVE Infrastructure	I	6	48	0	2500	2500
Cost Tool	S	1	40	1200	2000	3200
Risk Tool	S	1	40	1200	2000	3200
Assembly Sijmulation	S	1	40	1200	2000	3200
Factory Simulation	S	1	40	1200	2000	3200
Computerized Process Planner	S	1	40	1200	2000	3200
Tolerance Analysis	S	1	40	1200	2000	3200

	Category	# Heads	Manhours	Travel Exp	Course Cost	Total (Excludes MHrs)
Full Implementation Training Expenses						**$93,000**
SAVE Infrastructure	I	30	240	0	3000	3000
Cost Tool	S	5	200	0	15000	15000
Risk Tool	S	5	200	0	15000	15000
Assembly Sijmulation	S	5	200	0	15000	15000
Factory Simulation	S	5	200	0	15000	15000
Computerized Process Planner	S	5	200	0	15000	15000
Tolerance Analysis	S	5	200	0	15000	15000



Manhour Costs for Planning and Implementation

	Duration			% Alloc to		
	Days	**# People**	**% Time**	**Integration**	**Manhours**	**Total**
Reaching Decision to Implement						**$72,320**
Determine design teams that will use SAVE	15	6	0.1	1	72	7200
Determine approach to team members and subcontractors	15	2	0.5	1	120	12000
Select classes of simulation tools	10	6	0.05	0	24	2400
Select hardware platforms	10	2	0.05	0	8	800
Determine wrapping requirements for legacy tools	15	2	0.05	1	12	1200
Produce implementation plan	20	4	0.5	0.25	320	32000
Consider pilot exercise	13	6	0.05	0.25	31.2	3120
Produce business case - ROI	10	6	0.2	0.25	96	9600
Prepare for and brief maagement	5	2	0.5	0.25	40	4000
Decision to implement	0				0	0
Perform Pilot Exercise						**$647,440**
Indentify design problem with alternatives	15	2	0.5	0	120	12000
Select Tools	15	6	0.1	0	72	7200
Select hardware platforms	10	1	0.4	0	32	3200
Acquire software	22	1	0.2	0.15	35.2	3520
Acquire hardware	22	1	0.2	0.15	35.2	3520
Install system and test	22	xxx	xxx	0.2	280	28000
Train users - See estimates above	5	xxx	xxx	0.17	288	28800
Establish naming conventions for pilot team	15	6	0.1	1	72	7200
Develop or modify simulations and cost and risk models	30	6	0.5	0	720	72000
Gather data for design study - CAD models, mfg plans, etc	35	6	0.5	0	840	84000
Develop metrics measurement plans	22	1	0.25	0.25	44	4400
Execute design study	60	8	1	0	3840	384000
Measure metrics	50	6	0.04	0.25	96	9600
Report to management	3			0	0	0
Plan Full Scale Implementation						**$106,400**
Refine tool selection	15	6	0.5	0	360	36000
Validate hardware selection	10	2	0.05	0	8	800
Establish naming conventions for project / company	20	6	0.5	1	480	48000
Determine requirements for back-end data storage	20	3	0.25	1	120	12000
Plan phased release to design teams	15	8	0.1	0.25	96	9600
Report to management	1			0	0	0
Implement Full Scale System						**$528,800**
Acquire software	22	1	0.25	0.1	44	4400
Acquire hardware	22	1	0.25	0.1	44	4400
Install system and test	22	xxx	xxx	0.2	280	28000
Populate server with back-end data locations	15	xxx	xxx	1	1600	160000
Develop or modify simulaitons and cost and risk models	55	2	1	0	880	88000
Wrap legacy tools if required	55	xxx	xxx	1	1000	100000
Train users	20	xxx	xxx	0.17	1440	144000

Distribution Statement A: Approved for public release. Distribution is unlimited.

7.0 Implementation Benefits Estimation Spreadsheet

SAVE Benefits Estimation

Large Design Team Example

Primary Inputs

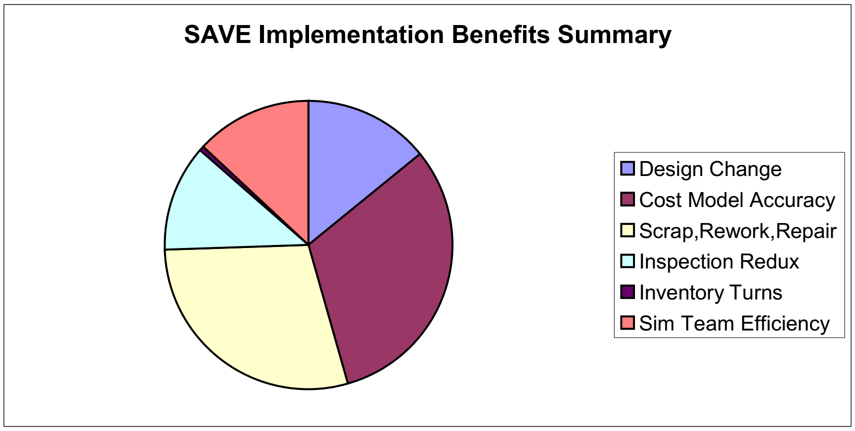
These inputs are used in multiple areas below.
Additional inputs are required in some areas.

🟨	Primary input fields
🟦	Secondary input fields
🟧	Calculated, but may be overwritten

Number of SAVE system users	30
Development Program Span - Years	5
Estimated Product Production Cost	$1,500,000
Number of Parts	1000
Number of Major Assemblies	4
Estimated Material Cost	$375,000
Estimated Fabrication Cost	$750,000
Estimated Assembly Cost	$375,000
Production rate per year	150
Number of units to count for savings	3000
Number of years to count for yearly savings	20
Sim Team efficiency increase due to SAVE	15.00%
Historical Error Rates/Cost	
Number of changes per part per year	0.15
Average cost per design change	25000
Historical % scrap, rework, repair	0.05

Summary Results

Benefits from simulation	**$74,381,250**
SAVE Benefits taken as Team Efficiency	**$4,320,000**
SAVE Benefits taken as increased simulation	**$11,157,188**
Total Benefits	**$85,538,438**



Distribution Statement A: Approved for public release. Distribution is unlimited.

Implementation Benefits Estimation Spreadsheet - Continued

Design Change Reduction

Number of changes per part per year	0.12	
Number of parts	1000	
Number of years considered	20	
Average cost per chage	$25,000	
SAVE Impact	0.2	F-22 Design Team estimate(reduced)
Estimated Savings	**$12,000,000**	

Design to Cost Accuracy

Number of units to count for savings	3000		
Number of design trade studies	20		
Approx value of approaches being traded	$50,000		
Percentage cost difference among approaches	15	Table Col Index	3
Traditional cost model accuracy	20	Table Row Index	4
Traditional percent correct decisions	78	(From chart)	
Cost of incorrect decisions - traditional	$99,000,000		
SAVE cost model accuracy	15	Table Row Index	3
SAVE percent correct decisions	84	(From chart)	
Cost of incorrect decisions - SAVE	$72,000,000		
Estimated Savings	**$27,000,000**		

Scrap, Rework, Repair Reduction

Unit production cost of product	$1,500,000	
Number of units	3000	
Historical percentage SR&R	0.05	
SAVE percentage impact	0.11	F-22 Design Team estimate
Estimated Savings	**$24,750,000**	

Fab & Assy Inspection Reduction

Cost of inspection as percent of labor cost	0.05	
Fab and Assy Cost per unit	$1,125,000	
Units per year	150	
Number of years considered	20	
SAVE impact percentage	0.06	F-22 Design Team estimate
Estimated Savings	**$10,125,000**	

Implementation Benefits Estimation Spreadsheet - Continued

Inventory Turn Increase

Per unit cost of product material and fab	$1,125,000	
Number of units in inventory	15	
Cost of material in inventory	$16,875,000	
Percent of material not being processed	0.25	Per Fentor, ManTech Roadmap Meeting
Yearly percentage inventory carrying cost	0.1	
Number of years to count	20	
Percent inventory scrapped by design change	0.05	
SAVE percentage impact	0.02	F-22 Design Team estimate
Estimated Savings	**$506,250**	

Mfg Simulation Team Efficiency

Number of SAVE system users	30	
Number of design years considered	5	
Average hourly cost	100	
Increase in efficiency from SAVE integration	15.00%	
Estimated Savings	**$4,320,000**	

8.0 Use of Metrics to Validate SAVE Benefits

8.1 Introduction

This section discusses a plan for metrics measurement and validation for a pilot-level implementation of SAVE-integrated manufacturing simulation tools. Overall metrics for the use of manufacturing simulation tools to support product development are discussed in Section 5.2 of this chapter.

Metrics measurement and validation for a small-scale design exercise is a challenging task. The statistical nature of the problems being addressed by manufacturing simulation makes conclusive metrics validation difficult. The problems are discussed and approaches to maximizing the benefits of performing a pilot exercise with SAVE are presented.

8.2 A Review of SAVE Metrics

The SAVE team has identified the following seven key metrics which virtual manufacturing technologies will impact.

- Design Change Reduction

  Simulation allows verification of designs, planning, processes and tools prior to making actual pasts thus producing better designs with fewer errors that require redesign.

- Scrap, Rework, Repair Reduction

  Simulation allows verification of designs, planning, processes and tools prior to making actual pasts thus producing better designs that have far fewer problems during production.

- Design to Cost Accuracy

  Accurate cost prediction methods allow better design choices to be made when performing trades between approaches which are close in cost.

- Lead Time Reduction

  Process optimization leads to better schedules and closer to just-in-time inventory.

- Process Capability

  Assembly tolerance buildup is simulated and yield predictions are developed. This yield information allows assembly process improvements that reduce variation to be identified.

- Fabrication & Assembly Inspection Reduction

Simulation allows optimization of the inspection process, produces designs with fewer errors to be found, and provides better shop floor training aides.

- Inventory Turn Increase

  Process simulation improves and reduces risk of just-in-time factory. Systems can be pulled rather than pushed in production leading to reduce work in progress.

The achievable cost savings will vary strongly with a project's size and complexity. The spreadsheet that is described above is available to help users make a quick estimate of the potential savings from five of these seven metrics plus effect of the productivity improvement of development team due to SAVE integration.

8.3 The Reality of Metrics Measurement and Validation

Accurate validation of the seven SAVE metrics within the scope of a pilot study is a challenging task. This is, in part, due to the statistical nature of the cost problem, and the relatively small number of design cases that can be included in a pilot scenario. For example, reduction of scrap, rework, and repair is a major contributor to SAVE-related savings. Not all designs require rework and when SAVE-supported designs reach production and no problems are found it will be difficult to validate the percentage of problems that were avoided. Conclusive validation can only come from larger statistical samples. Similarly, the improved design to cost accuracy provided by SAVE provides a higher percentage of correct design decisions. Many correct decisions are made with traditional methods, and the best choice will not always be found due to imprecision in cost methods.

SAVE metrics validation is a challenge, but certainly not impossible. The plan presented below has been developed within the above constraints and will maximize the information within the limited statistical sample of a small-scale SAVE pilot.

8.4 Metrics Measurement During a SAVE Pilot

The SAVE pilot team should document a number of the elements of their activities while the design study is underway to provide data upon which to base metrics validation. Each of these elements is identified below and their use in metrics is discussed.

8.4.1 Record Cost, Schedule, and Risk Estimates Made through Simulation

These are the primary trade study values that are generated by the SAVE system. The SAVE Data Model captures these estimates as simulation runs are made, and they will be summarized in the electronic design notebook. These values will be compared to actual values measured on the shop floor as products are produced (see discussion below). The team will document the accuracy of estimates derived from simulation and will identify the sources of inaccuracy. Wherever possible these comparisons will be made at the operation or job step level.

8.4.2 Document Problems, Design Errors, and Design Improvements Discovered through Simulations

The basic premise of virtual manufacturing and SAVE is that problems found during design are much less expensive to fix than those found during production. The process of designing complex systems is one of discovering the "best" approach to a product or process. History has proven again and again that many problems will escape even the best designers as they create complex products to tight schedules on limited budgets. The ultimate success of SAVE will be measured through the systematic reduction of errors on a statistically large sample of designs. On the small sample of designs represented by SAVE's demonstrations and pilot tests it may be difficult to accurately measure successful metrics (although we are just as likely to over-demonstrate a metric as under-demonstrate one). Recording every case where simulation helps a designer discover a better approach or eliminate a problem will increase the sample size to improve confidence in the benefits of the SAVE system. Each case of a simulation-based improvement must be carefully scrutinized to assess whether the traditional design process would have reached the same conclusion during design, on the shop floor, or not at all.

8.4.3 Track Time Spent in Utilizing SAVE System

SAVE pilot scenarios are actually run over an extended period of weeks or months. During this time, the SAVE pilot team should record their time spent on the various activities related to the design study. Recorded times should be categorized as follows:

- Data preparation
- Model building
- Data input
- Execution
- Data output
- Analysis
- Design team coordination
- Reporting

These measurements will allow the team to assess the benefits of the integration framework provided by SAVE and will provide the foundation for estimating implementation and use costs for the intended production program.

8.4.4 SAVE Metrics Validation Using Production Results

The ideal pilot exercise will involve a design activity that is scheduled to go into production quickly. Final validation of the benefits of manufacturing simulation can truly only be measured by improvements in the manufacturing process. Even with less than an ideal pilot case, it is possible to get a meaningful measure of the advantages of integrated manufacturing simulation tools. The plan presented here will address both cases, where the SAVE pilot design goes to production and the worst case, where it does not, or is at least delayed.

In the course of performing the pilot design and simulating the manufacturing processes, problems with either the design or the processes will be uncovered. These, of course, will be remedied immediately but they should be documented as indications of the types of problems that might have been discovered on the shop floor. Estimates of the cost to remedy these problems with traditional approaches should be made and will become part of SAVE validation results.

In general, metrics validation will be accomplished by comparing the results of the SAVE simulations, as described above, to comparable measurements made on the shop floor. These comparisons will be made for both cost and schedule estimates at both process plan and individual operation/job step levels. Comparisons will be used to validate the individual simulation tools and will also point out where improvements in the tools and their integration should be made.

There is, of course, the possibility that some problem will be missed in the simulations and will be discovered in production. This will be documented and investigated to identify whether a change is needed in the use of a particular simulation tool, or in the overall concept of operations of the SAVE system.

One of the major benefits of the use of SAVE is the potential for a significant improvement in the time to build the first article and in the learning curve. Complete validation of these effects will only come from a statistically significant number of tests, but the pilot team should measure the data and begin the comparison process. Learning curve data, in particular, will take time to develop and to show improvements over historical values.

8.4.5 SAVE Metrics Validation without Using Production Results

If the designs created by the SAVE pilot exercise are not adopted for production, or production is not near-term, validation of SAVE metrics will be much more difficult. Some validation can still be accomplished as described below.

Without production results the pilot team will be forced to rely on simulation to simulation comparisons. If it is assumed that the baseline design is what would have gone to production in the traditional process, the cost, schedule, and risk improvements identified through simulation can be attributed to the SAVE simulation tools.

8.5 Summary and Conclusions

The SAVE development team recognizes that measurement and validation of affordability-related metrics is a vital element of successful implementation at many sites. While this plan discusses the potential difficulty in obtaining this objective, the SAVE team remains confident that the benefits are real and achievable, can be proven in a pilot implementation, and can be fully validated as SAVE is applied to larger scale design activities. As SAVE is implemented at other sites, these results can also be reviewed to help develop the case for wider-spread the use of integrated manufacturing simulation tools.

Chapter 5

User’s Guide for SAVE-Developed Software

SAVE Software User’s Manual
Contract Number F33615-95-C-5538
CDRL A012

CHAPTER 5 – TABLE OF CONTENTS

1.0 Available User Documentation

User documentation was developed for each component of the SAVE environment. This documentation describes the functions and capabilities of the SAVE-compliant wrappers for the commercial manufacturing simulation tools. In addition, there are user-level manuals for each of the SAVE team-developed software components.

This documentation is contained in appendices to this document. Each component user guide is in a separate appendix as follows:

Appendix D – ASURE Wrapper User's Guide

Appendix E – Cost Advantage Wrapper User's Guide

Appendix F – Factor AIM Wrapper User's Guide

Appendix G – IGRIP and QUEST Wrapper User's Guide

Appendix H – VSA-3D Wrapper User's Guide

Appendix I – CATIA Costlink User's Guide

Appendix J – Cost Model User's Guide

Appendix K – Project 98 Wrapper User's Guide

Appendix L – Work Flow Manager User's Guide

Appendix M – Query Manager User's Guide

It is intended that, in production implementations of the SAVE Virtual Manufacturing Environment, commercial versions of these software packages and their associated user documentation will be obtained from the appropriate commercial software vendors. With that in mind, these User Guides are provided as reference material only.

In addition to the user guides, the SAVE team developed training material for each component of the system. This training material, provided in the form of a PowerPoint presentation, is included in Appendix N.

Appendix A

Sample SAVE Usage Scenarios

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

1.0 Typical SAVE Scenario

In Chapter 2, there is a discussion of a typical SAVE usage scenario in somewhat general terms. It includes steps for requirements flowdown, data gathering, concept analysis, concept selection, and tool usage. The best way, however, to gain a true understanding of the use of SAVE is to review an actual example. This appendix will discuss three specific examples of the use of the SAVE Virtual Manufacturing Environment. The first example, the Initial Demonstration, was conducted early in the program at the end of Phase I. The second example, the Interim Demonstration, was conducted midway through Phase II with about one year left in the program. The third example describes the SAVE Final Demonstration conducted at the end of the contracted effort.

2.0 Initial Demonstration – F-16 Horizontal Stabilizer

The phase one SAVE demonstration involved the redesign of the F-16 horizontal stabilizer. This original design modification actually occurred in the early 80's; however, the events associated with the change provide an excellent example of how the SAVE tool suite would be used in an IPT setting. The original F-16 horizontal stabilizer was a honeycomb core bonded panel assembly that underwent an engineering redesign to increase the stabilizer area by 20%. The results of stress and weight analysis were sufficient to rule out an increased area honeycomb core bonded panel assembly early in the design evaluation. For the purposes of the SAVE demonstration, the actual corrugated spar construction and a hypothetical rib spar design were used to develop assembly process trades, manufacturing process refinements, and detail part trades. Figure A-1 provides an overview of the overall decision process and final selection of the corrugated spar assembly.

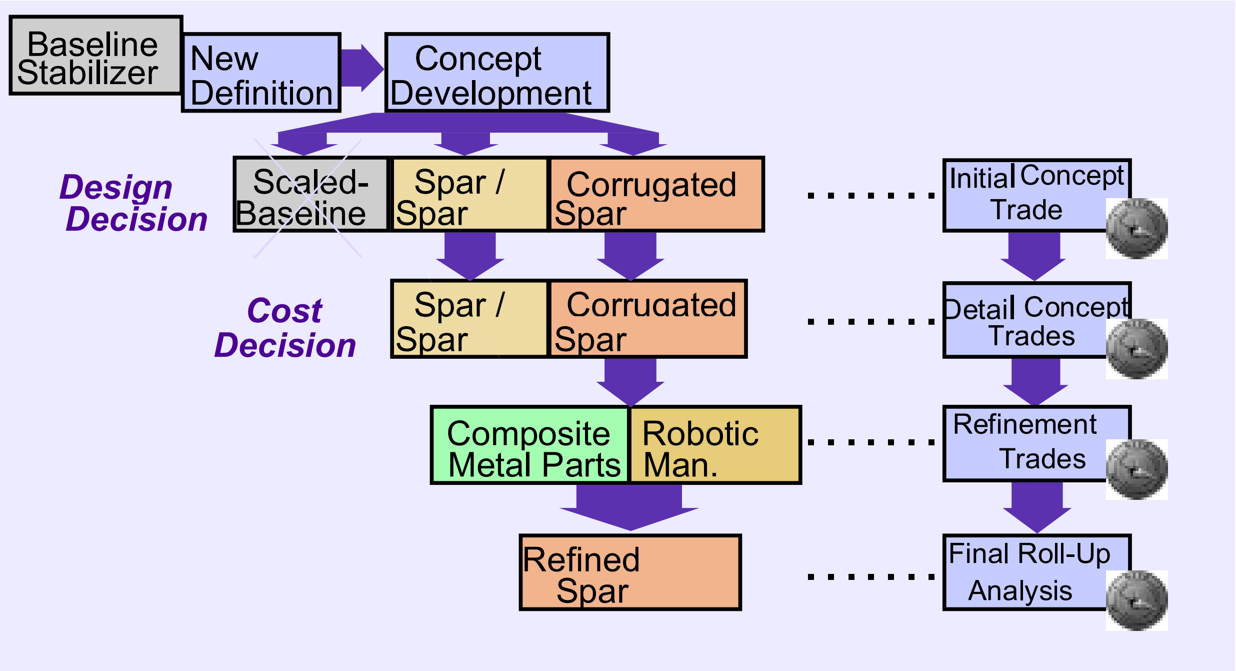
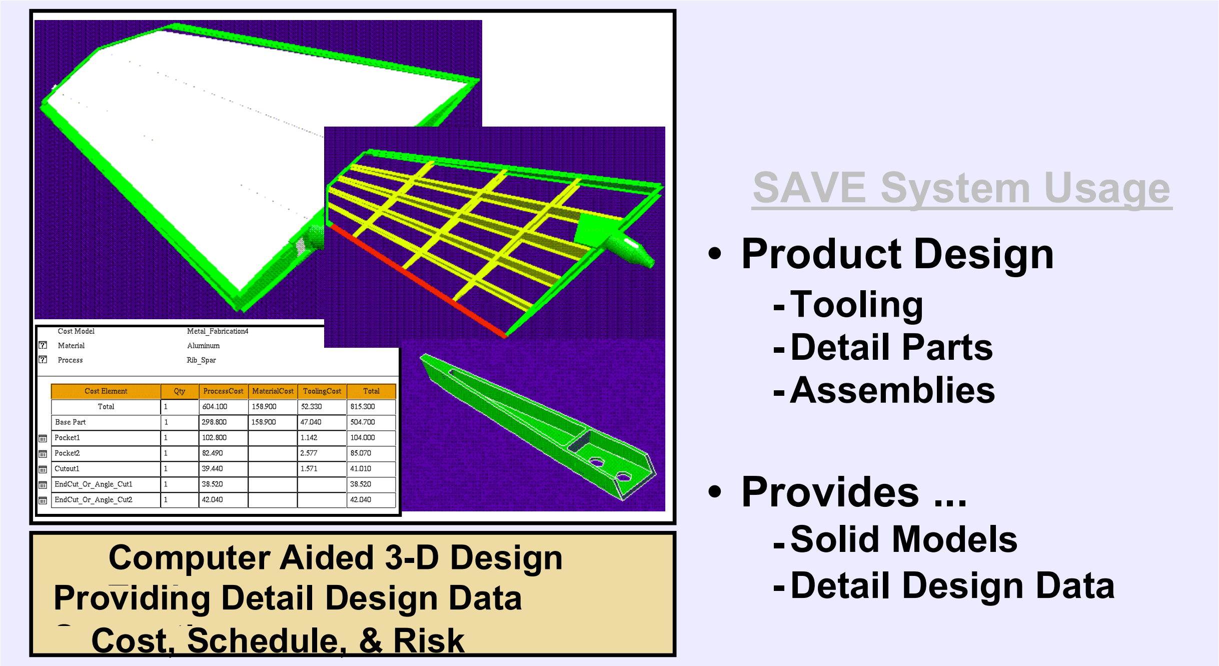


Figure A-1: Overview of the Decision Process

2.1 Structural Concept Selection

During the structural concept selection activity, three candidate designs were proposed: a scaled up version of the original honey comb core bonded panel assembly, a rib spar design with attached composite skins, and a sheet metal corrugated spar design with attached composite skins. Figure A-2 presents the CAD tool used to model these designs.



Cost Model Metal_Fabrication4

Material Aluminum

Process Rib_Spar

Cost Element	Qty	ProcessCost	MaterialCost	ToolingCost	Total
Total	1	604.100	158.900	52.330	815.300
Base Part	1	298.800	158.900	47.040	504.700
Pocket1	1	102.800		1.142	104.000
Pocket2	1	82.490		2.577	85.070
Cutout1	1	39.440		1.571	41.010
EndCut_Or_Angle_Cut1	1	38.520			38.520
EndCut_Or_Angle_Cut2	1	42.040			42.040

Computer Aided 3-D Design
Providing Detail Design Data
Cost, Schedule, & Risk

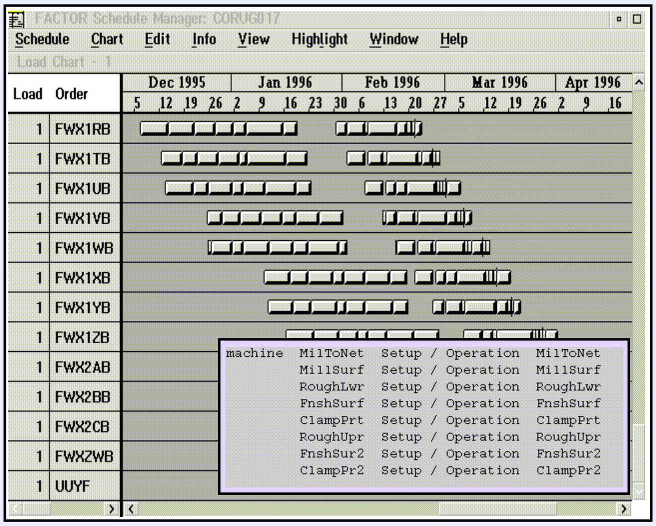
SAVE System Usage

- Product Design
  - Tooling
  - Detail Parts
  - Assemblies
- Provides ...
  - Solid Models
  - Detail Design Data

Figure A-2: CATIA Used to Model Three Structural Concepts

As mentioned above, the engineering stress analysis results were sufficient to rule out a scaled up version of the original honeycomb core bonded panel assembly early in the design process. Subsequently, the two remaining alternatives were given preliminary process plans and evaluated in terms of cost, schedule, and risk. Tools used to perform this task include ergonomics of the manual assembly operations, discrete event simulation to determine process times resource requirements, and overall span, and cost assessment to determine the cost for both options. In this comparison using manual assembly techniques, cost and schedule were the main factors for choosing the corrugated spar over the rib spar configuration (structural concept selection) since the risk would be comparable for both options when using similar assembly fixtures, manual drilling, and fastening techniques. The preliminary simulation results indicated that:

- The rib spar design would require more assembly fixtures and assembly labor than the corrugated spar design to meet schedule span requirements (Figure A-3).
- The rib spar design would require more detail components and associated detail fabrication costs.



Load	Order
1	FWX1RB
1	FWX1TB
1	FWX1UB
1	FWX1VB
1	FWX1WB
1	FWX1XB
1	FWX1YB
1	FWX1ZB
1	FWX2AB
1	FWX2BB
1	FWX2CB
1	FWXZWB
1	UUYF

```
machine  MilToNet  Setup / Operation  MilToNet
         MillSurf  Setup / Operation  MillSurf
         RoughLwr  Setup / Operation  RoughLwr
         FnshSurf  Setup / Operation  FnshSurf
         ClampPrt  Setup / Operation  ClampPrt
         RoughUpr  Setup / Operation  RoughUpr
         FnshSur2  Setup / Operation  FnshSur2
         ClampPr2  Setup / Operation  ClampPr2
```

Discrete Event Schedule Simulations Allowing Accurate Determination Of Span Time And Resource Requirements

PRITSKER

SAVE System Usage

- **Schedule Assessment**
  - **Parts Fabrication**
  - **Assembly Operations**
- **Work Flow Optimization**
- **Provides ...**
  - **Span Times**
  - **Resource Requirements**
  - **Process Plans**

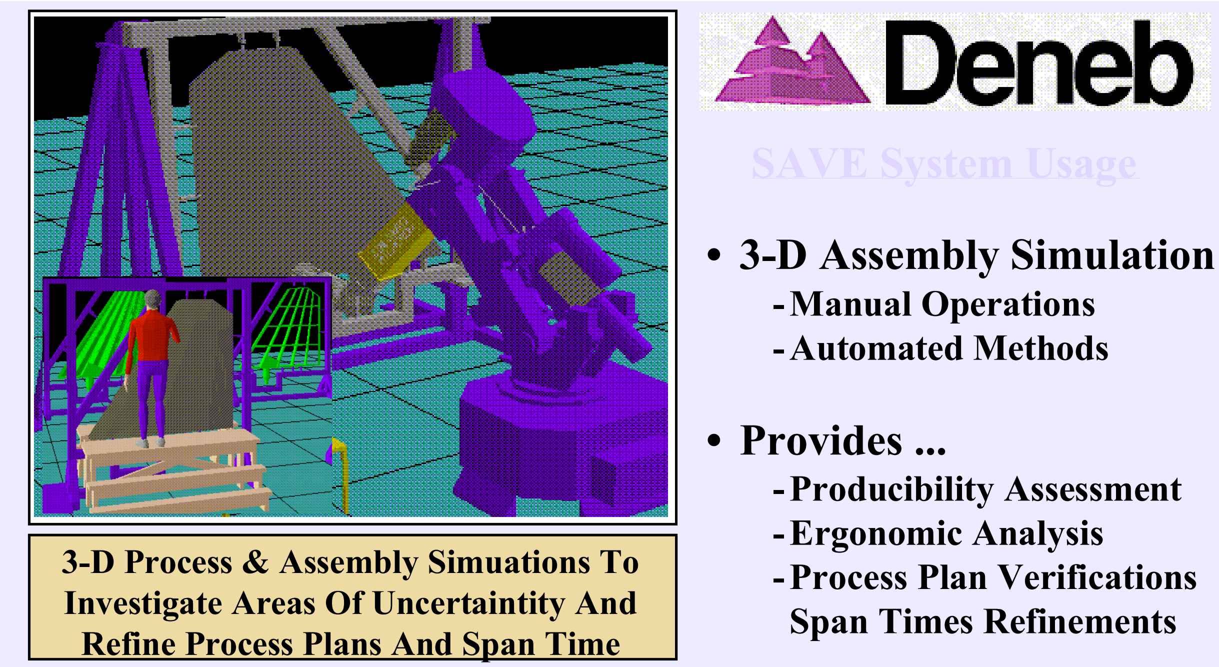
Figure A-3: Schedule Analysis for Two Remaining Concepts

2.2 Manufacturing Method Trades

Once the corrugated version was selected, manufacturing assembly plan modifications including robotic drilling were considered. The drilling options were evaluated by comparing ergonomic analysis of the manual drilling process to IGRIP analysis of the robotic drilling process. After comparing the results of the two simulations, a significant reduction in span time for the composite skin drilling/countersink operation was indicated for the robotic drilling process. Additionally, risk assessment of manual versus robotic drilling/countersinking of the composite skins indicated that significantly more rework would result if the manual drilling process were used. Figure A-4 presents a screen shot of the assembly cell evaluation tool used for the manufacturing method trades.

In summary, cost and risk were the primary factors for selecting robot drilling over manual drilling for the composite skin attachment process (manufacturing method trades). The findings were as follows:

- Robot drilling provides an overall reduction in cycle time for the drilling operation thus reducing cost.
- Robot drilling provides a much smaller variance with respect to the nominal countersink depth requirements which reduces the need for fastener and surface rework (milling & filling) as compared to the manual drilling process.



Deneb

SAVE System Usage

- **3-D Assembly Simulation**
  - **Manual Operations**
  - **Automated Methods**

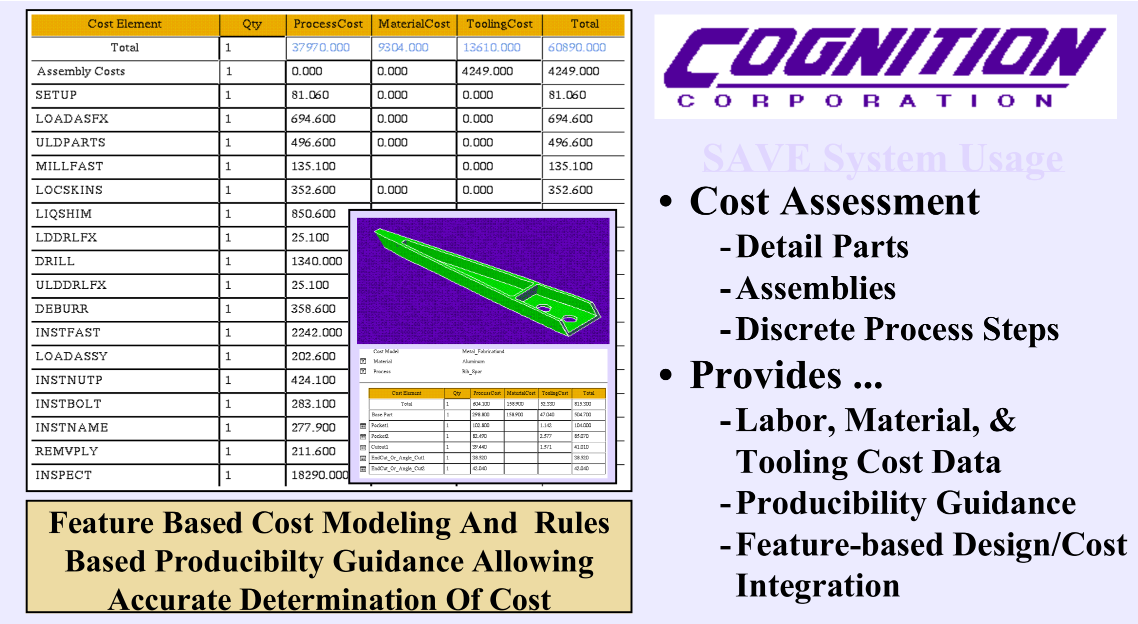
- **Provides ...**
  - **Producibility Assessment**
  - **Ergonomic Analysis**
  - **Process Plan Verifications**
  Span Times Refinements

3-D Process & Assembly Simuations To Investigate Areas Of Uncertaintity And Refine Process Plans And Span Time

Figure A-4: Ergonomics/IGRIP Assembly Cell Simulation

2.3 Detail Part Trades

Once the corrugated version was selected, detail part trades were performed on various components of the horizontal stabilizer assembly. The assembly of the horizontal stabilizer requires the attachment of a sub assembly (leading edge assembly) to the stabilizer during the final assembly process steps. This sub assembly is a bonded panel design and a material compatibility problem with one of the baseline components (machined root rib) and the leading edge sub assembly was anticipated. In this instance risk was the driving factor. No schedule impact was indicated, however additional cost was estimated by the subsequent cost assessment. A machined aluminum root rib is less expensive to fabricate than a composite root rib, but potential material compatibility problems with the next assembly justified the use of the composite root rib for this application. Figure A-5 presents a screen shot for the detail component cost analysis for the leading edge root rib.



Cost Element	Qty	ProcessCost	MaterialCost	ToolingCost	Total
Total	1	37970.000	9304.000	13610.000	60890.000
Assembly Costs	1	0.000	0.000	4249.000	4249.000
SETUP	1	81.060	0.000	0.000	81.060
LOADASFX	1	694.600	0.000	0.000	694.600
ULDPARTS	1	496.600	0.000	0.000	496.600
MILLFAST	1	135.100		0.000	135.100
LOCSKINS	1	352.600	0.000	0.000	352.600
LIQSHIM	1	850.600			
LDDRLFX	1	25.100			
DRILL	1	1340.000			
ULDDRLFX	1	25.100			
DEBURR	1	358.600			
INSTFAST	1	2242.000			
LOADASSY	1	202.600			
INSTNUTP	1	424.100			
INSTBOLT	1	283.100			
INSTNAME	1	277.900			
REMVPLY	1	211.600			
INSPECT	1	18290.000			

Cost Model	Metal_Fabrication4
Material	Aluminum
Process	Rib_Spar

Cost Element	Qty	ProcessCost	MaterialCost	ToolingCost	Total
Total	1	604.100	158.900	52.330	815.300
Base Part	1	298.800	158.900	47.040	504.700
Pocket1	1	102.800		1.142	104.000
Pocket2	1	82.490		2.577	85.070
Cutout1	1	39.440		1.571	41.010
EndCut_Or_Angle_Cut1	1	38.520			38.520
EndCut_Or_Angle_Cut2	1	42.040			42.040

Feature Based Cost Modeling And Rules Based Producibilty Guidance Allowing Accurate Determination Of Cost

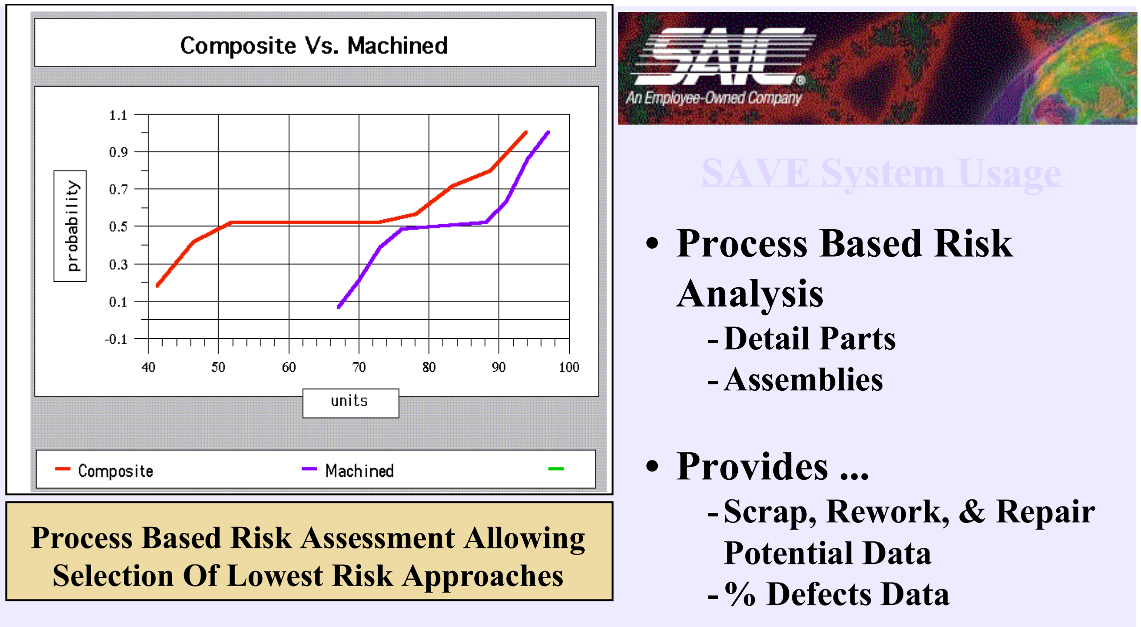
COGNITION CORPORATION

SAVE System Usage

- **Cost Assessment**
  - Detail Parts
  - Assemblies
  - Discrete Process Steps
- **Provides ...**
  - Labor, Material, & Tooling Cost Data
  - Producibility Guidance
  - Feature-based Design/Cost Integration

Figure A-5: Detail Component Cost Analysis

Figure A-6 presents a screen shot for the detail component risk analysis for the leading edge root rib.



Process Based Risk Assessment Allowing Selection Of Lowest Risk Approaches

SAVE System Usage

- **Process Based Risk Analysis**
  - Detail Parts
  - Assemblies
- **Provides ...**
  - Scrap, Rework, & Repair Potential Data
  - % Defects Data

Figure A-6: Detail Component Risk Analysis

3.0 Interim Demonstration – Redesign of the F-22 Gunport

The SAVE Interim Demonstration involved a redesign activity on the F-22 gunport. Initial concept testing showed that the blast from the gun was eroding the composite skin and surrounding structure. The design team identified three options that might resolve the problem – metallic skin, split metallic/composite skin or replaceable trough insert with skin cover. Two of the alternatives were discarded for performance-related issues; therefore, the SAVE team performed manufacturing simulations and additional trade studies for the insert/cover alternative. Typically, a team will evaluate all options at some level, narrow the choices, and perform more detailed evaluations of those alternatives. The ability to share common data afforded by the integrated SAVE environment enables the IPPT to quickly evaluate many scenarios with respect to their relative cost, schedule or risk.

The evaluation of the proposed gunport design concept involved two primary areas of analysis. The first was an assessment of the gunport assembly. The purpose of this analysis was to determine the optimal assembly plan including factory layouts, tooling and labor requirements. The requirement for this assembly process plan was that it must meet the required F-22 rates and schedule while minimizing impacts to cost and risk. The second assessment involved a decision between titanium and stainless steel for the cover material. The goal of this analysis was to determine if there were clear drivers for this decision in either the cost or risk area. Figure A-7 shows the process flow that was defined by the IPPT for the gunport trade studies evaluated using SAVE.

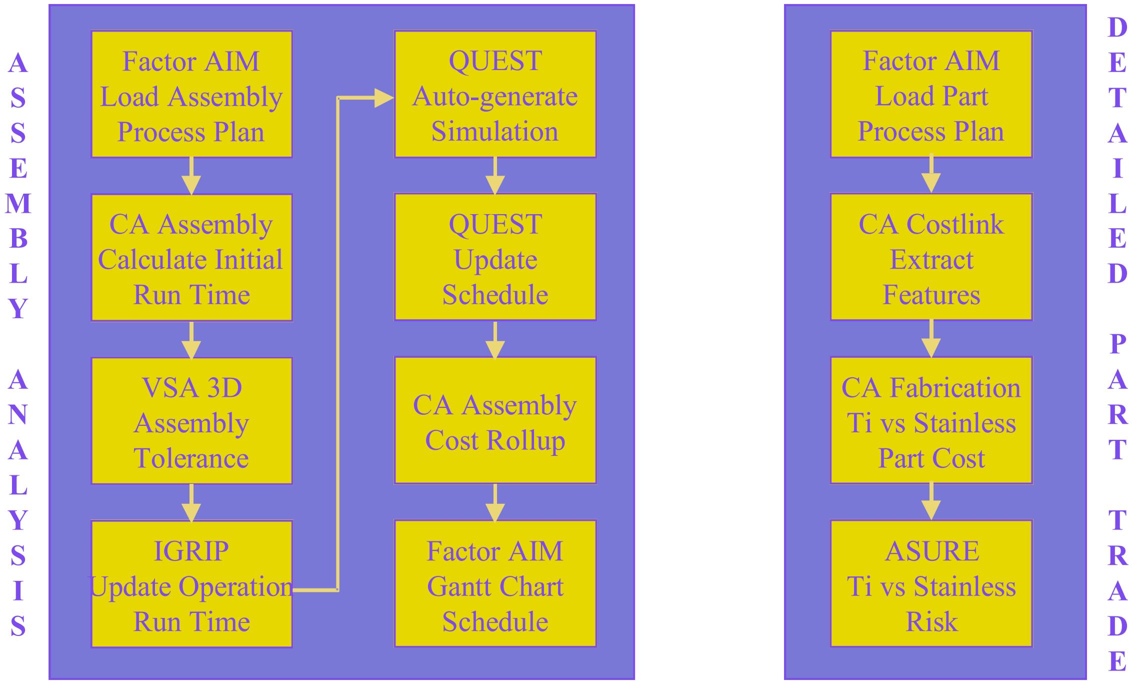

ASSEMBLY ANALYSIS
Factor AIM Load Assembly Process Plan
CA Assembly Calculate Initial Run Time
VSA 3D Assembly Tolerance
IGRIP Update Operation Run Time
QUEST Auto-generate Simulation
QUEST Update Schedule
CA Assembly Cost Rollup
Factor AIM Gantt Chart Schedule
Factor AIM Load Part Process Plan
CA Costlink Extract Features
CA Fabrication Ti vs Stainless Part Cost
ASURE Ti vs Stainless Risk
DETAILED PART TRADE

Figure A-7 Interim Demonstration Process Flow

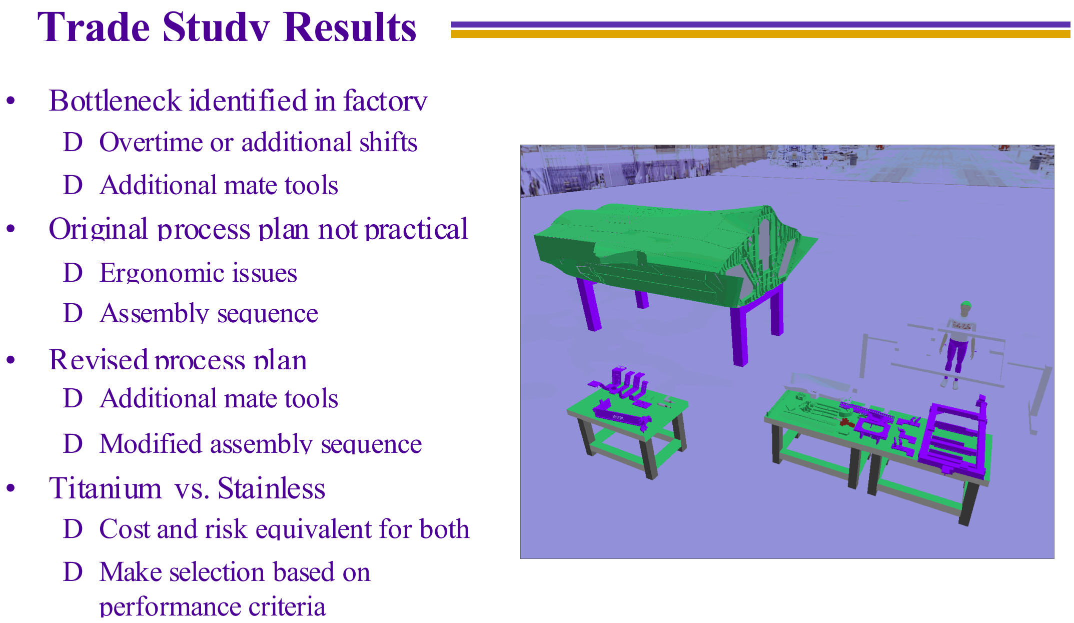
With the knowledge of the assessment goals, the IPPT selected the appropriate SAVE tools for each analysis. In determining the order for executing the simulations, the team considered several factors. Since the primary benefits of SAVE come from the tool integration, the team wanted to maximize the amount of data available while minimizing the data entry time. The spreadsheet-like interface of Factor AIM provided a quick and easy method for entering the major elements of the process plan while maximizing the data available to the downstream tools. Table A-2 shows the primary elements of the SAVE data model that were shared by the simulation tools. This list is not all-inclusive, but it does provide a general idea of the types of information that are useful as a starting point.

Table A-2: Examples of SAVE Common Data for Interim Demonstration

	Factor AIM	CA Assy	VSA 3D	IGRIP	QUEST	CA Fab	ASURE
Process Plan							
Name	X	X	X	X	X	X	X
Description	X	X	X	X	X	X	X
Operation							
Name	X	X	X	X	X	X	X
Description	X	X	X	X	X	X	X
Precedents	X	X	X	X	X	X	X
Run Time	X	X		X	X		
Type	X	X	X	X	X	X	X
Resource App							
Name	X			X	X		
Description	X			X	X		
Quantity	X			X	X		
Resource Pool							
Name	X			X	X		
Description	X			X	X		
Quantity	X			X	X		
Personnel							
Name	X			X	X		
Description	X			X	X		
Skill	X			X	X		
Tool							
Name	X			X	X		
Description	X			X	X		
Type	X			X	X		
CAD Model							
Name				X	X	X	
Description				X	X	X	
Location				X	X	X	
Schedule							
Actual Start Date	X				X		
Actual End Date	X				X		
Actual Duration	X				X		

Once the initial tool was selected, the team ordered the tools based on their ability to provide input to downstream tools as well as the level of fidelity of their estimates. The initial run times were calculated by the knowledge bases in Cost Advantage. These times served as a good starting point for conducting the detailed simulations. Risk related to tolerance buildup was important for refining the assembly sequence prior to simulating the assembly workcells. With detailed simulations in IGRIP, the team was able to update the run times for specific operations within the plan that were of particular interest. In addition, some ergonomic issues were identified and addressed. For example, one drilling operation required an ambidextrous operator to accomplish the task. A re-sequencing of the steps eliminated this requirement. The QUEST simulation model was automatically generated from data that was exported by the other tools into the SAVE database. It included the parts, tools, locations, and assembly sequence for the gunport. The QUEST simulation made modifications to the sequence and resource requirements based on identified bottlenecks and other problem areas. This updated plan was used by Cost Advantage for the final cost assessment. Once the updates were complete, the process plan was imported back into Factor AIM for final schedule analysis relative to the F-22 requirements.

For the detailed part trade, the team was evaluating the fabrication process for the proposed cover. Once again, Factor AIM provided the starting point with the fabrication process plan. The CATIA Costlink was used to extract the cover feature data for use by the Cost Advantage knowledge base. The relative cost of the titanium versus the stainless part was determined by Cost Advantage using the extracted feature data and the imported process plan. The process plan was also imported into ASURE and was used along with SPC data to evaluate the relative risk of the two options. The results of these assessments showed that cost and risk were virtually equal for the titanium and stainless steel covers. This allowed the IPPT to make their decision based on other performance-related criteria. Figure A-8 summarizes the results of the gunport trade studies with the primary findings in the area of assembly sequencing pictured on the right.



Trade Study Results

- Bottleneck identified in factory
  - D Overtime or additional shifts
  - D Additional mate tools
- Original process plan not practical
  - D Ergonomic issues
  - D Assembly sequence
- Revised process plan
  - D Additional mate tools
  - D Modified assembly sequence
- Titanium vs. Stainless
  - D Cost and risk equivalent for both
  - D Make selection based on performance criteria

Figure A-8: Gunport Trade Study Results

4.0 Final Demonstration – F-22 Weapons Bay Doors

The SAVE Final Demonstration focused on an actual problem area that was being addressed by the F-22 program – interference and mismatch in the main weapons bay door installation. Figure A-9 shows the F-22 midbody with the three doors and the skin involved in the fit problem. The primary area of interference was occurring between the auxiliary seal door and the skin shown in Figure A-10. This problem was compounded by the fact that the doors and midbody were manufactured in one location and installed in another. The lag time in feedback on installation and fit problems made problem resolution difficult.

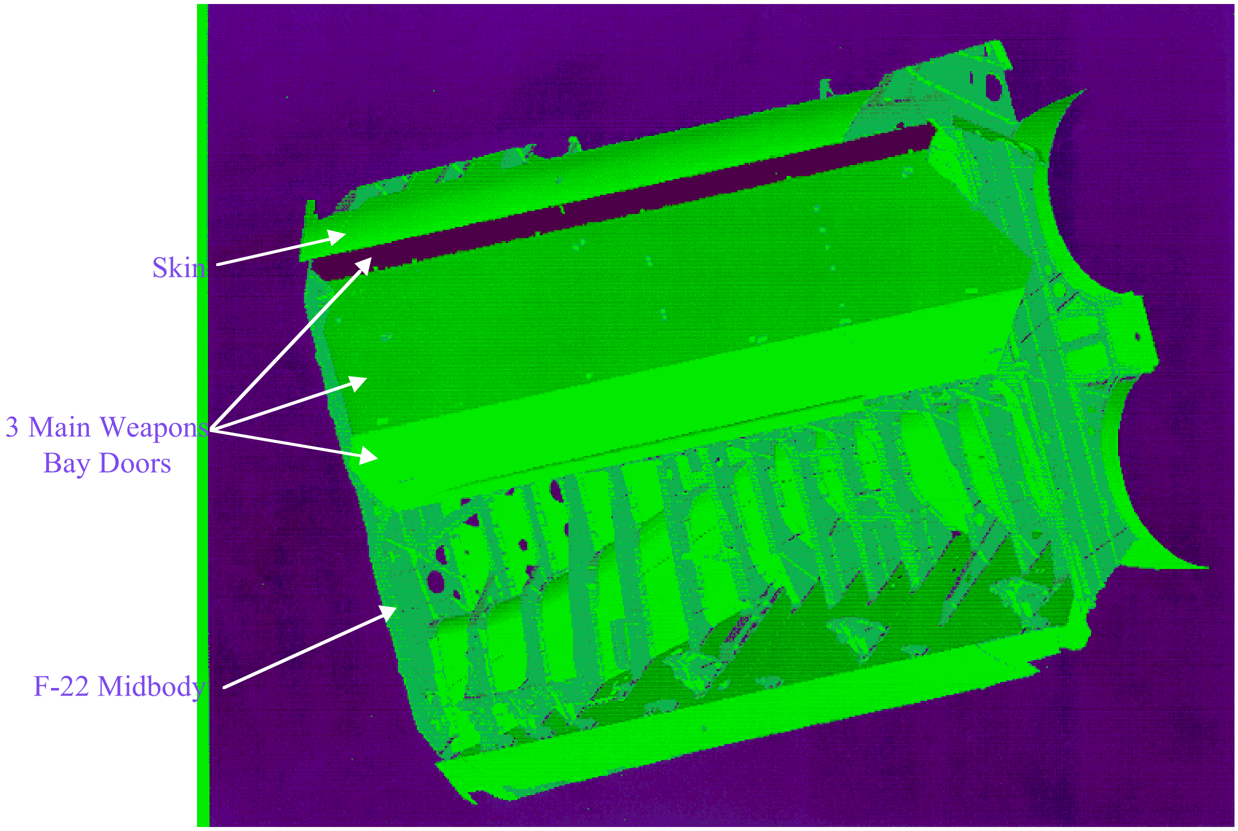


Figure A-9: F-22 Midbody with Doors

As part of this demonstration, the SAVE team used the integrated virtual manufacturing environment to evaluate two options that were being explored by the F-22 program. The first study evaluated the effect of a change in the tooling concept, whereas the second study addressed the addition of a fit check process prior to component shipment. The team used IGRIP to visualize the changes that were being proposed and to determine the appropriate areas for further simulation.

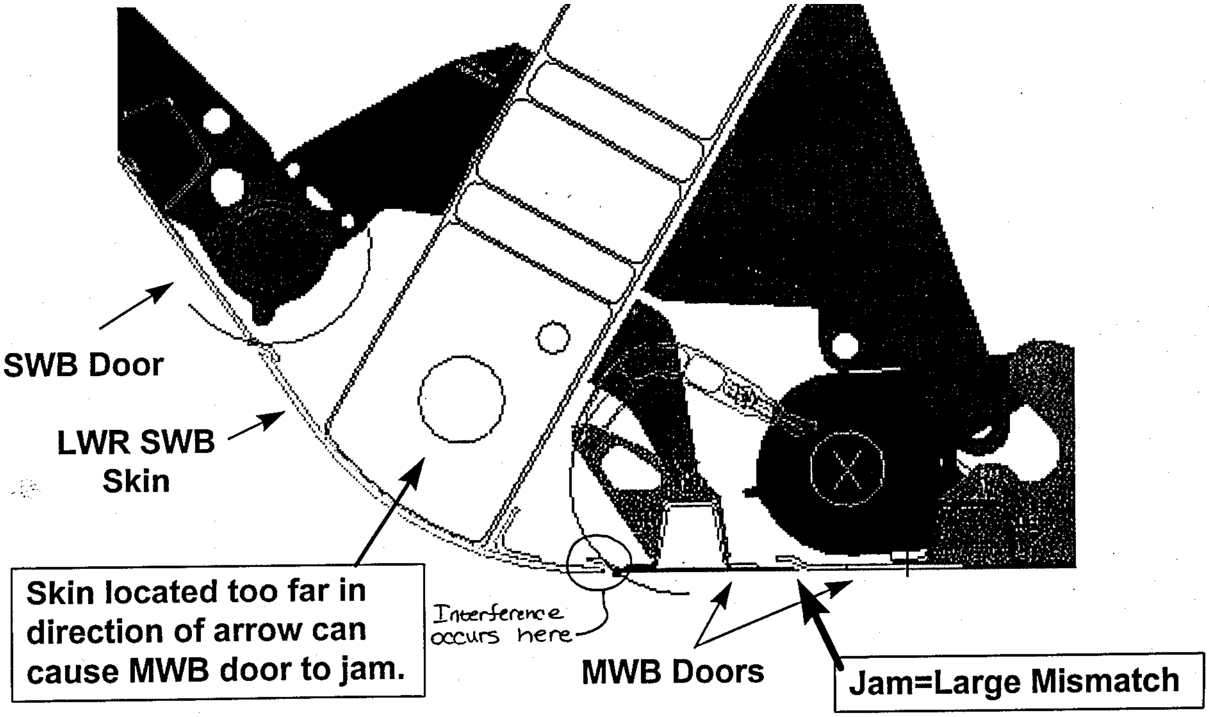


Figure A-10: Door to Skin Interference

4.1 Tooling Trade Study

The tooling study evaluated the effect of changing from an inner mold line (OML) to an outer mold line (OML) tooling philosophy. Simulations were used to determine if holding the OML would produce a better fit between the skin and the doors. Figure A-11 shows the process used for this evaluation.

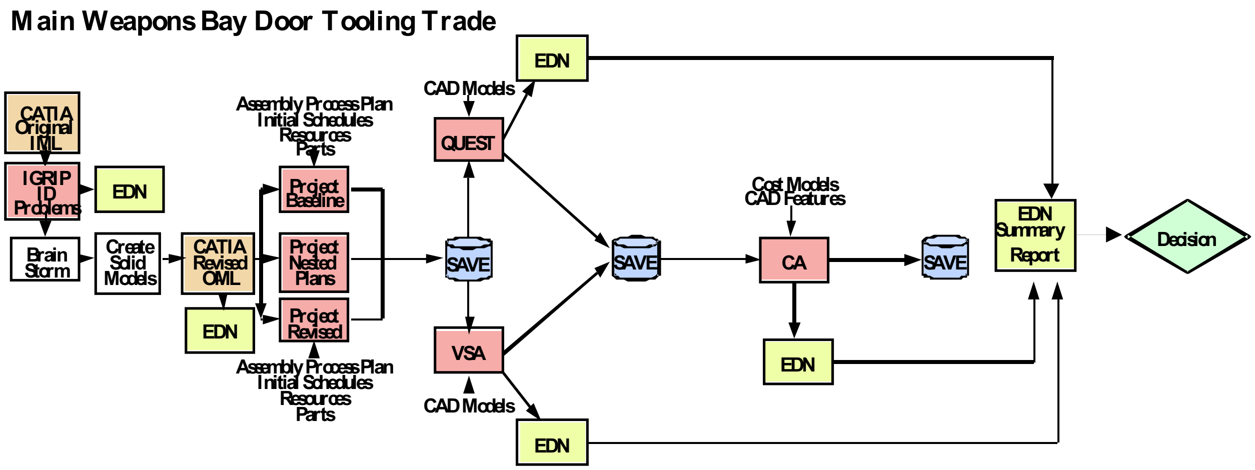
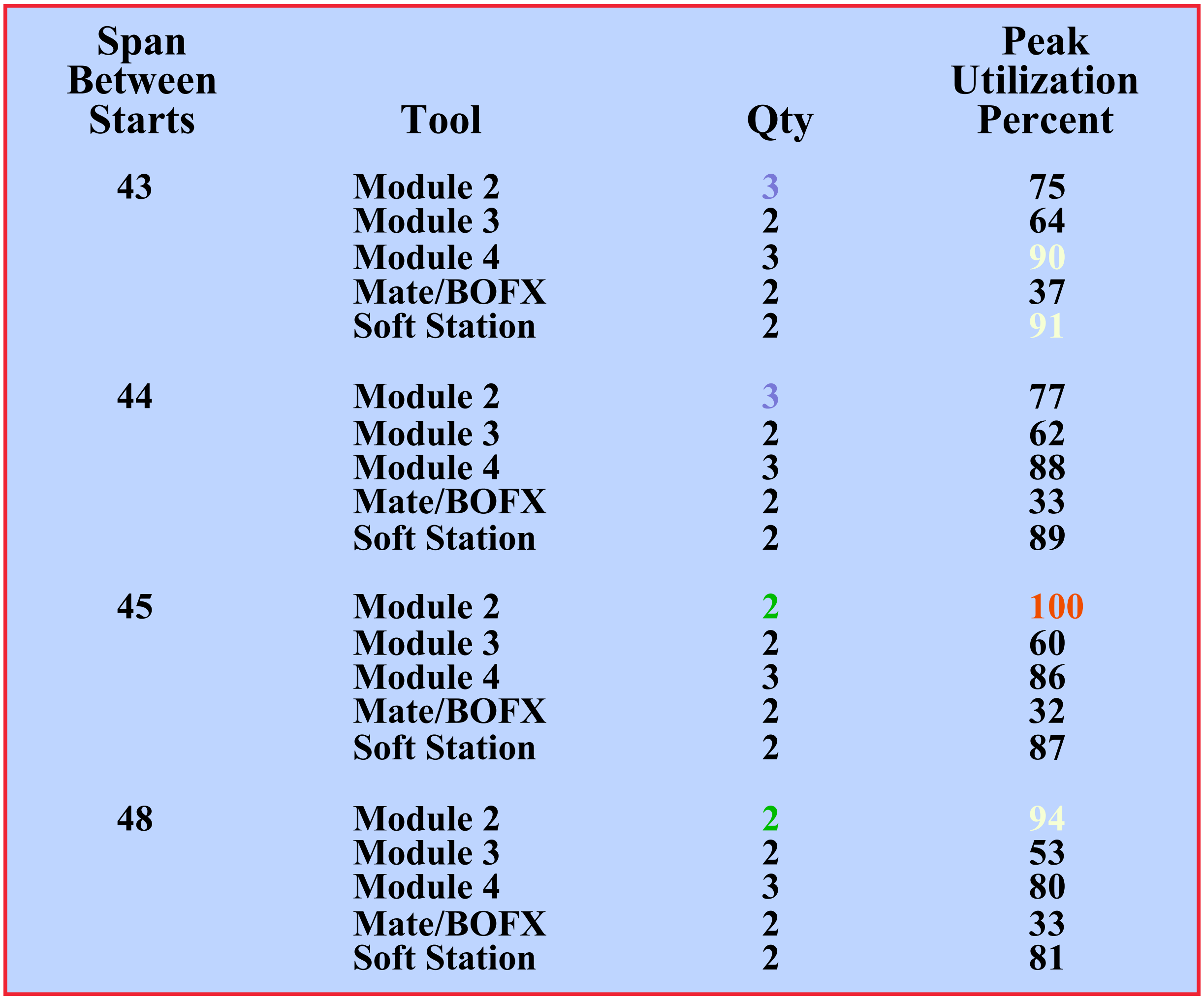


Figure A-11: Tooling Trade Flow Diagram

The Manufacturing Engineer (ME) within the team used Microsoft Project to develop the initial process plans for the tooling trade. Since this trade involved a modification of an existing F-22 process, the original plan was imported into Project and used as a starting point for the process planning activity. The resultant plan included several levels of indenture as appropriate for the different types of simulations that were performed.

QUEST was used to perform an overall rate tooling analysis for the midbody assembly process. The highest level planning information that included the steps in the assembly process, the tooling, parts, their locations, and the process durations were imported from SAVE. The simulation showed an unacceptable utilization of one of the tools in the assembly. An internal trade was conducted that varied span times, number of tools, and crew size to determine the optimum solution to the over-utilized tooling. Figure A-12 tabulates the results. The most attractive option was the addition of one module two jig as shown in Figure A-13.



Span Between Starts	Tool	Qty	Peak Utilization Percent
43	Module 2	3	75
	Module 3	2	64
	Module 4	3	90
	Mate/BOFX	2	37
	Soft Station	2	91
44	Module 2	3	77
	Module 3	2	62
	Module 4	3	88
	Mate/BOFX	2	33
	Soft Station	2	89
45	Module 2	2	100
	Module 3	2	60
	Module 4	3	86
	Mate/BOFX	2	32
	Soft Station	2	87
48	Module 2	2	94
	Module 3	2	53
	Module 4	3	80
	Mate/BOFX	2	33
	Soft Station	2	81

Figure A-12: Rate Tooling Trade Study Results

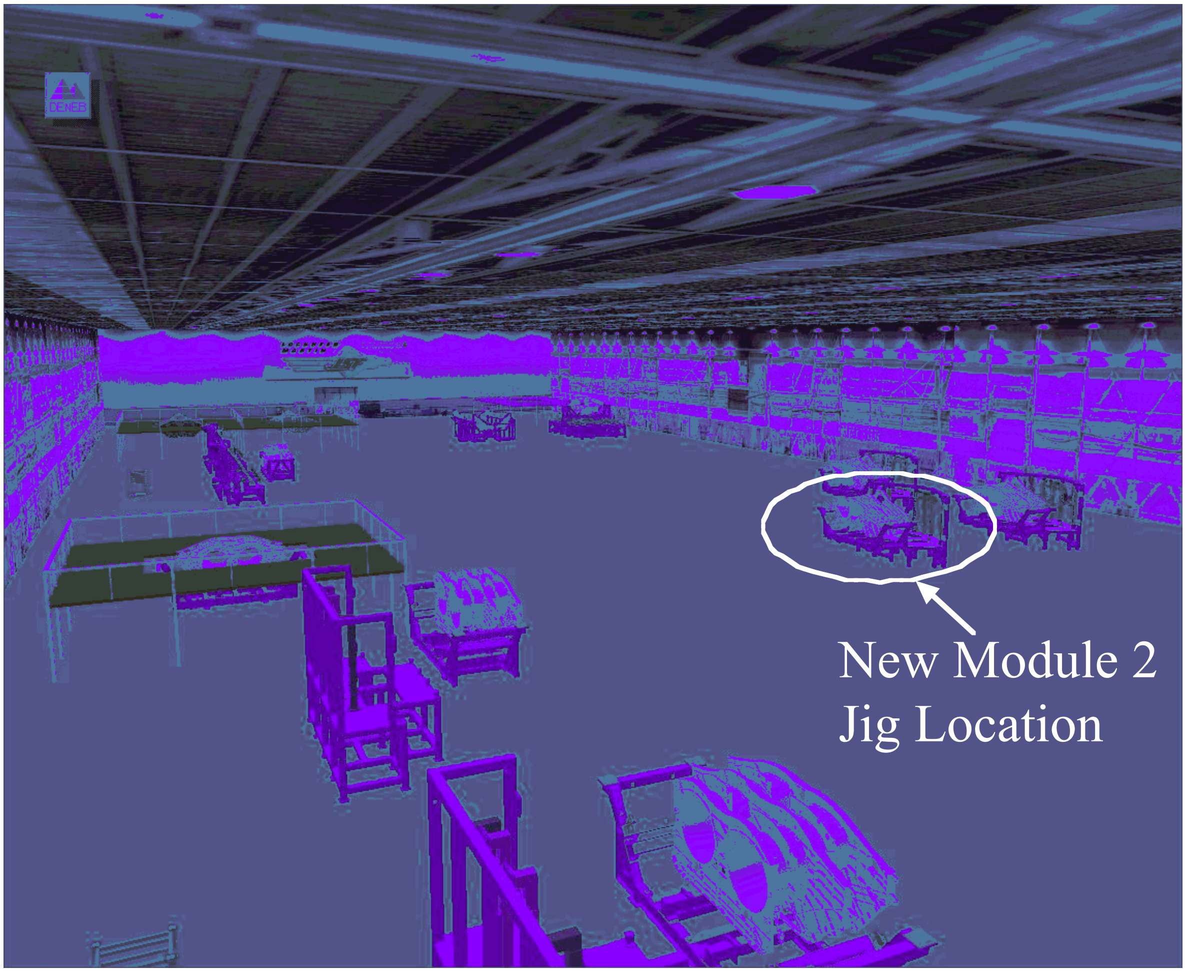
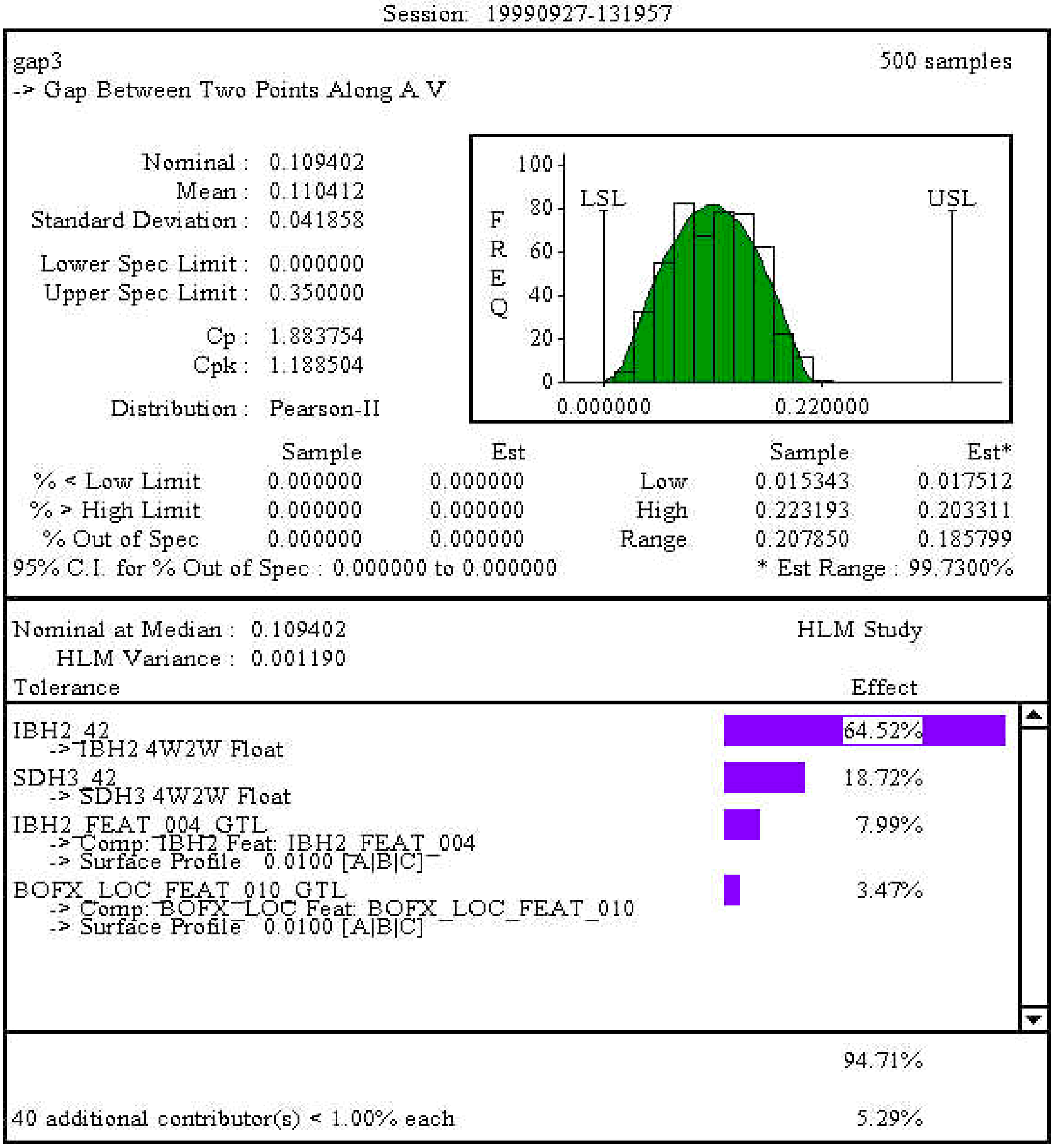


Figure A-13: QUEST Model of the Proposed Concept

Using the detailed assembly sequence information for the auxiliary seal door and the permanent skin from SAVE, the VSA tool performed a tolerance analysis that analyzed the details of the door-to-skin mismatch. The SAVE process planning information was melded with the dimension and tolerance data stored in the CAD model to create the tolerance simulation model. The simulation results showed that the OML tooling change eliminated most, but not all of the fit problems. The remaining contributors to the mismatch were identified as two tooling holes. By modifying the tooling pin diameter, the final fit problems were eliminated and the process made repeatable. Figure A-14 shows a results comparison from the tolerance simulation.

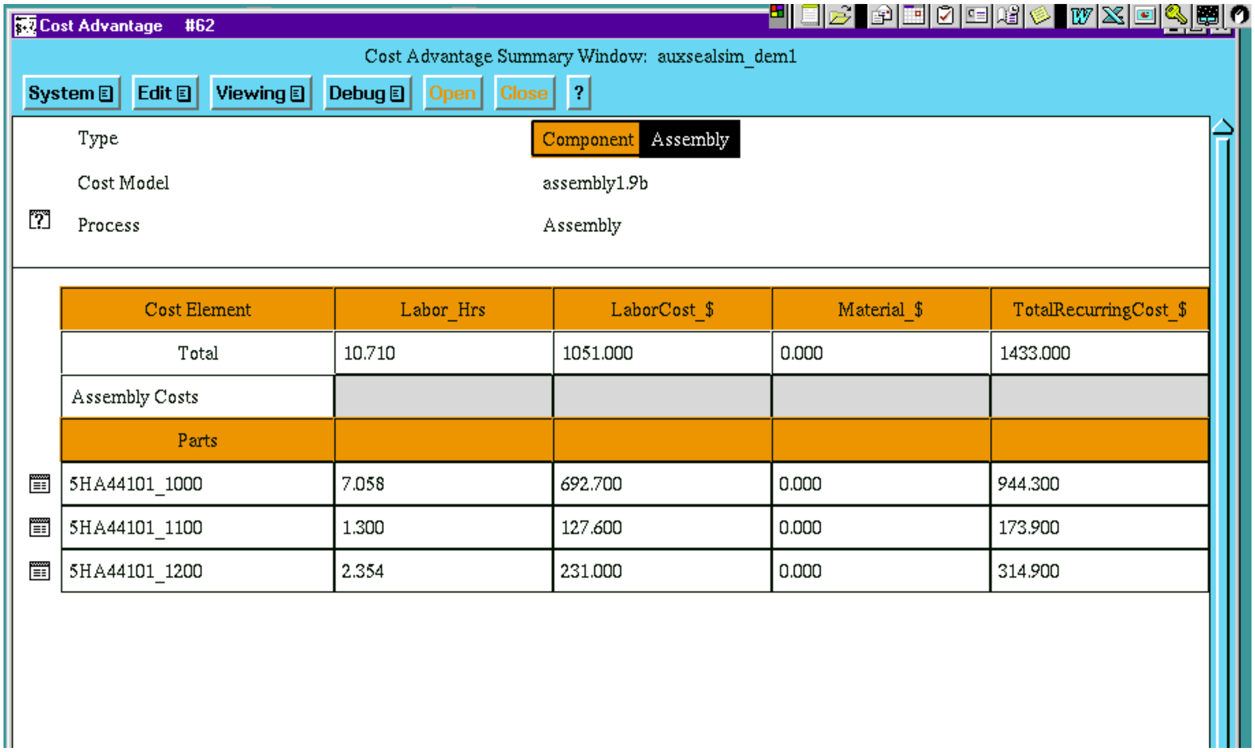
The Cost Advantage knowledge-based cost assessment tool used feature data extracted from the CATIA CAD model for the auxiliary seal door with the process planning information extracted from SAVE to make an overall recurring cost estimate for the assembly process using the new tooling concept. This analysis used the assembly cost model that was one of four cost models developed under the SAVE contract. Figure A-15 shows the results of this analysis.



```
                              Session:  19990927-131957
gap3                                                                      500 samples
-> Gap Between Two Points Along A V

              Nominal :  0.109402
                 Mean :  0.110412
   Standard Deviation :  0.041858

    Lower Spec Limit :  0.000000
    Upper Spec Limit :  0.350000

                  Cp :  1.883754
                 Cpk :  1.188504

        Distribution :  Pearson-II

                        Sample          Est                       Sample        Est*
   % < Low Limit        0.000000        0.000000        Low       0.015343      0.017512
   % > High Limit       0.000000        0.000000        High      0.223193      0.203311
    % Out of Spec       0.000000        0.000000        Range     0.207850      0.185799
95% C.I. for % Out of Spec : 0.000000 to 0.000000                 * Est Range : 99.7300%

Nominal at Median :  0.109402                                   HLM Study
     HLM Variance :  0.001190
Tolerance                                                           Effect

IBH2_42                                                             64.52%
    -> IBH2 4W2W Float
SDH3_42                                                             18.72%
    -> SDH3 4W2W Float
IBH2_FEAT_004_GTL                                                    7.99%
    -> Comp: IBH2 Feat: IBH2_FEAT_004
    -> Surface Profile   0.0100 [A|B|C]
BOFX_LOC_FEAT_010_GTL                                                3.47%
    -> Comp: BOFX_LOC Feat: BOFX_LOC_FEAT_010
    -> Surface Profile   0.0100 [A|B|C]

                                                                    94.71%

40 additional contributor(s) < 1.00% each                            5.29%
```

Figure A-14: Tolerance Results

The Virtual Manufacturing environment predicted positive results for making the tooling change. The mismatch and fit problems are largely eliminated by incorporating the OML tooling concept and modifying the geometry of two tooling holes. The costs for implementing this change were acceptable when accounting for the cost incurred as a result of the rework caused by the fit problems.



Cost Advantage Summary Window: auxsealsim_dem1

Type	Component Assembly
Cost Model	assembly1.9b
Process	Assembly

Cost Element	Labor_Hrs	LaborCost_$	Material_$	TotalRecurringCost_$
Total	10.710	1051.000	0.000	1433.000
Assembly Costs				
Parts				
5HA44101_1000	7.058	692.700	0.000	944.300
5HA44101_1100	1.300	127.600	0.000	173.900
5HA44101_1200	2.354	231.000	0.000	314.900

Figure A-15: Cost Analysis for Auxiliary Seal Door Assembly

4.2 Fit Check Trade Study

The fit check study evaluated adding a process to install the main weapons bay doors in the midbody to check for any fit or mismatch problems prior to their shipment to the final assembly facility. Simulations were used to determine the effect, particularly to schedule and cost, of adding this fit check. Figure A-16 shows the simulation process flow for this evaluation.

Once again, Microsoft Project was used for the initial process planning activity. Information available from the final assembly process was imported into project and used as a starting point for the planning.

Factor AIM was used to evaluate the effect of this change of the rate tooling requirements for the F-22. The plan including the operations, labor and tooling requirements were imported from SAVE. The simulation was conducted with the current F-22 rate requirement of eight soft stations. The fit check was simulated along with the existing processes that take place in that tool. There were two key finding as a result of the simulation. First there was very little impact to the F-22 schedule with the fit check addition. The checks took between 8 and 10 hours to complete and could be accomplished within the allotted calendar time for midbody and door

Main Weapons Bay Door Fit Check Trade

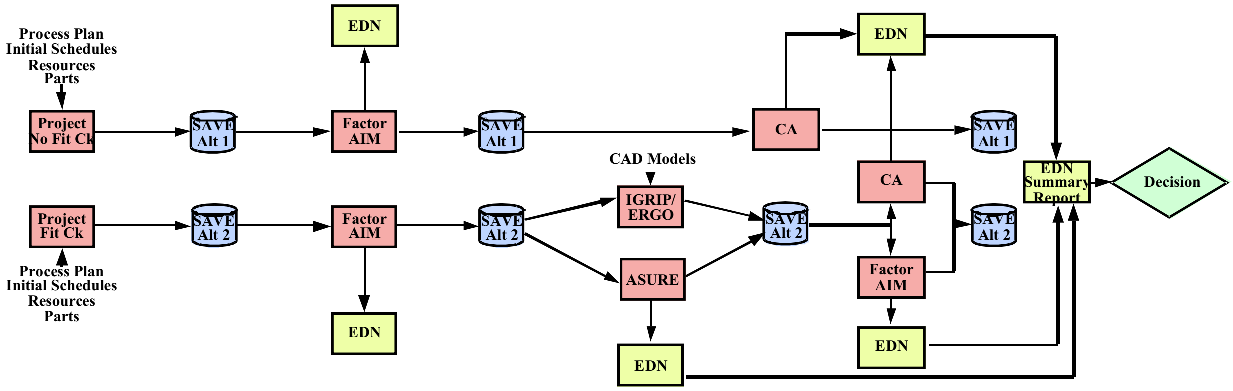


Figure A-16: Fit Check Process Flowchart

shipment. In addition, the check was accomplished with the existing tooling with any impact to the other processes that took place in the tool. Figure A-17 summarizes the tool utilization both with and without the fit check.

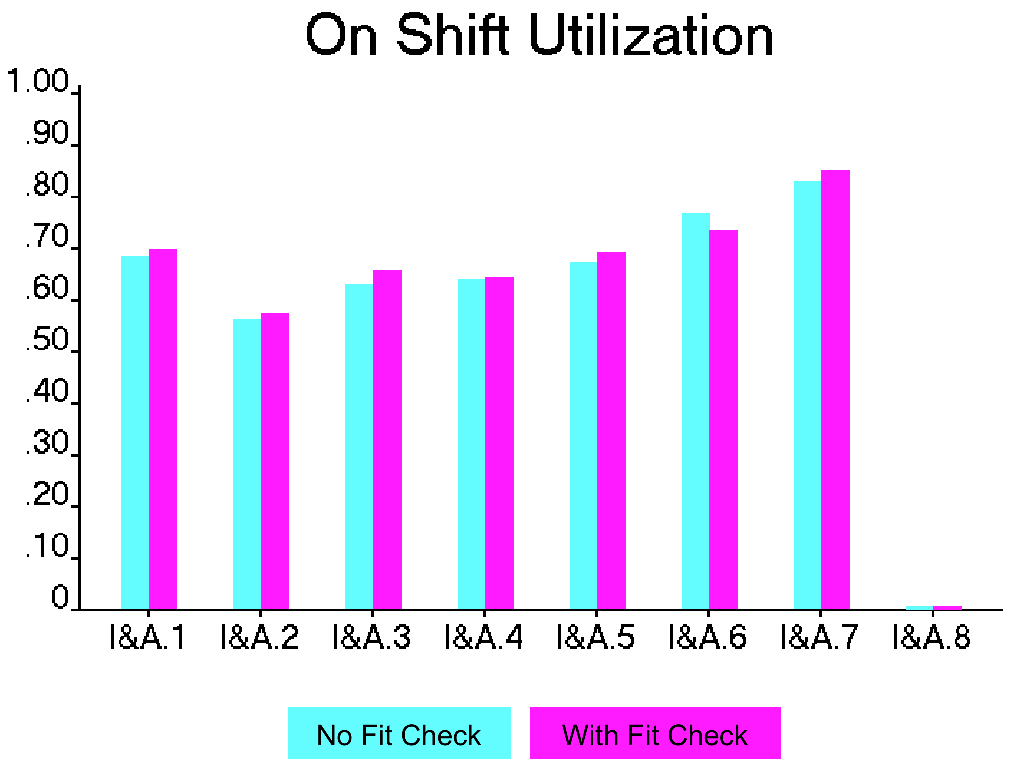
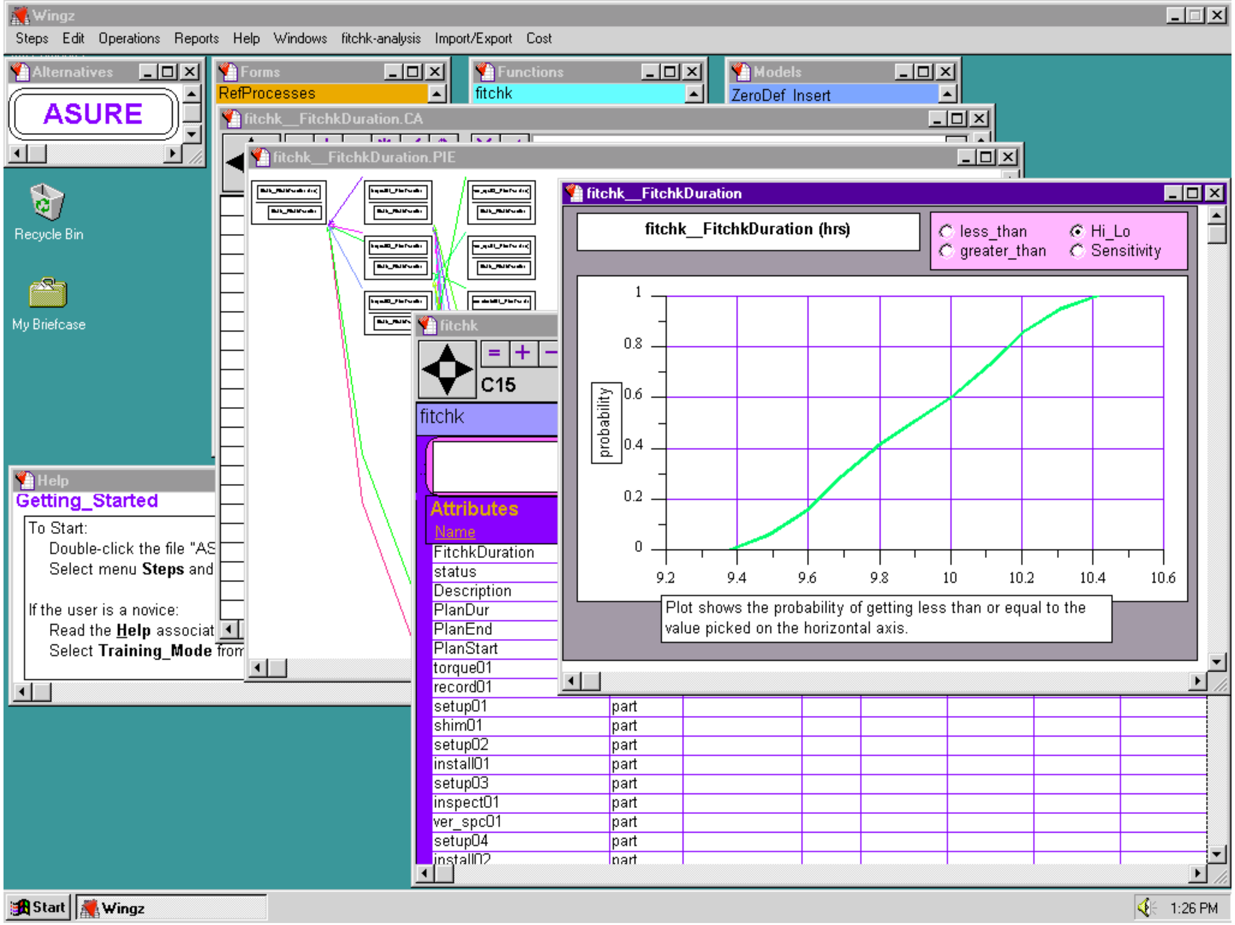


Figure A-17: Tool Utilization

ASURE was used to evaluate the schedule risk associated with adding the fit check. The tool read the process plan with the Factor AIM-updated schedule times and tooling requirements from SAVE. The analysis showed the probability of success for a range of schedule values. These results are graphed in Figure A-18.



Getting_Started

To Start:
Double-click the file "AS
Select menu **Steps** and

If the user is a novice:
Read the **Help** associat
Select **Training_Mode** from

fitchk__FitchkDuration (hrs)

○ less_than ● Hi_Lo
○ greater_than ○ Sensitivity

Plot shows the probability of getting less than or equal to the value picked on the horizontal axis.

Name	
FitchkDuration	
status	
Description	
PlanDur	
PlanEnd	
PlanStart	
torque01	
record01	
setup01	part
shim01	part
setup02	part
install01	part
setup03	part
inspect01	part
ver_spc01	part
setup04	part
install02	part

Figure A-18: Probability of Success versus Process Time for Fit Check.

Ergonomic assessments were made using the ERGO tool's simulation model that was created from the process plan, tooling, personnel, part, and location data imported from SAVE. A trade study was performed to determine the number of people required to perform the fit check process. The numbers of people were varied from three to five with simulation run at each value. The trade resulted in five people required – four to hold the door in place and one to install the pins. Door weight and personnel stances required to hold the door in place were factors in the result.

Cost assessments were made for the addition of the fit check. Cost Advantage used the actual simulation results for task duration and personnel requirements to estimate the cost. The total recurring cost for the fit check was XXX.

Simulation showed positive results for adding the fit check. There was no additional tooling required and there was minimal impact to the schedule. Recurring costs were not negligible, but the fit check would only be necessary until the process is proven and repeatable.

4.3 Results

These two virtual manufacturing studies provided useful feedback to the F-22 design team that was performing similar trade studies in parallel. The OML tooling change eliminated the majority of the fit and mismatch problems. In addition, incorporating a fit check process provided the opportunity to fix problems in a timely manner, thus reducing the cost of extensive rework downstream.

The integrated environment played an important part in the effective use of the virtual manufacturing tools for these trade studies. There was extensive data reuse in this trade study. The process planning information including all operations, parts and resources were created once in the planning tool and used without user intervention in the other simulation tools. Data generated by the simulations was also shared among the tools for a synergistic affect on cost, schedule and risk estimates. Since the data needed by the simulation models was available through the SAVE shared environment, the model generation times were reduced by 20-50 percent depending upon the extent of data sharing and the capability of the tool wrapper.

Appendix B

Use Cases

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

This appendix contains use cases that were developed for submittal to the Object Management Group (OMG) as a response to their Manufacturing Domain Task Force Request for Information #4.

Use Case: Manufacturing Simulation In Support of Design Study

Overview:

This use case describes the role that a set of integrated manufacturing simulation tools can play in the design of a mechanical product and its manufacturing processes. The use of these tools is built around the widely applied concept of an Integrated Product Team (IPT) organization. The set of tools is selected to support assessment of the cost, schedule, and risk impacts of both product and process design decisions. Design of complex products always involves compromise, and a balanced understanding of all impacts of design alternatives is necessary for a successful, affordable product. Many design disciplines have developed extensive simulation and analysis capability to support their positions in a design compromise discussion. Finite Element Models clearly and graphically demonstrate product strength inadequacies. Rapid, accurate weight analysis based on solid CAD models unequivocally demonstrates weight problems. Computational Fluid Dynamics can now accurately predict complex flows to highlight potential performance or control problems. Even highly subjective issues such as styling are being quantified through accurate customer surveying techniques. Historically, manufacturing producibility and cost issues have been more subjective and can be forced to take second place to more clearly demonstrated problems when compromise in a design is required.

Over the past few years excellent commercial manufacturing simulation tools have become available that allow manufacturing considerations to be equally weighed with other design disciplines. These tools include shop floor cell simulations, including robotics and ergonomic models, discrete event factory flow, accurate feature and process-based cost models, 3-D tolerance analysis for complex assemblies, and advanced risk analysis that rigorously track risk from uncertain inputs to final results. These tools all help produce the manufacturing plan and share in its data. Many of these tools can utilize data from other tools to improve the accuracy of their simulations. For instance, accurately simulated times and resource requirements can lead to more accurate cost estimates. Integrated data sharing among these tools is vital to their efficient, timely use in a fast-paced design environment. Figure B-1 shows the roles of the IPT members in the use of manufacturing simulation tools.

The scenario below will describe the overall team actions required for the use of an integrated set of manufacturing simulation tools in a design trade study. Use of the individual tools within the study will be discussed in the other use cases.

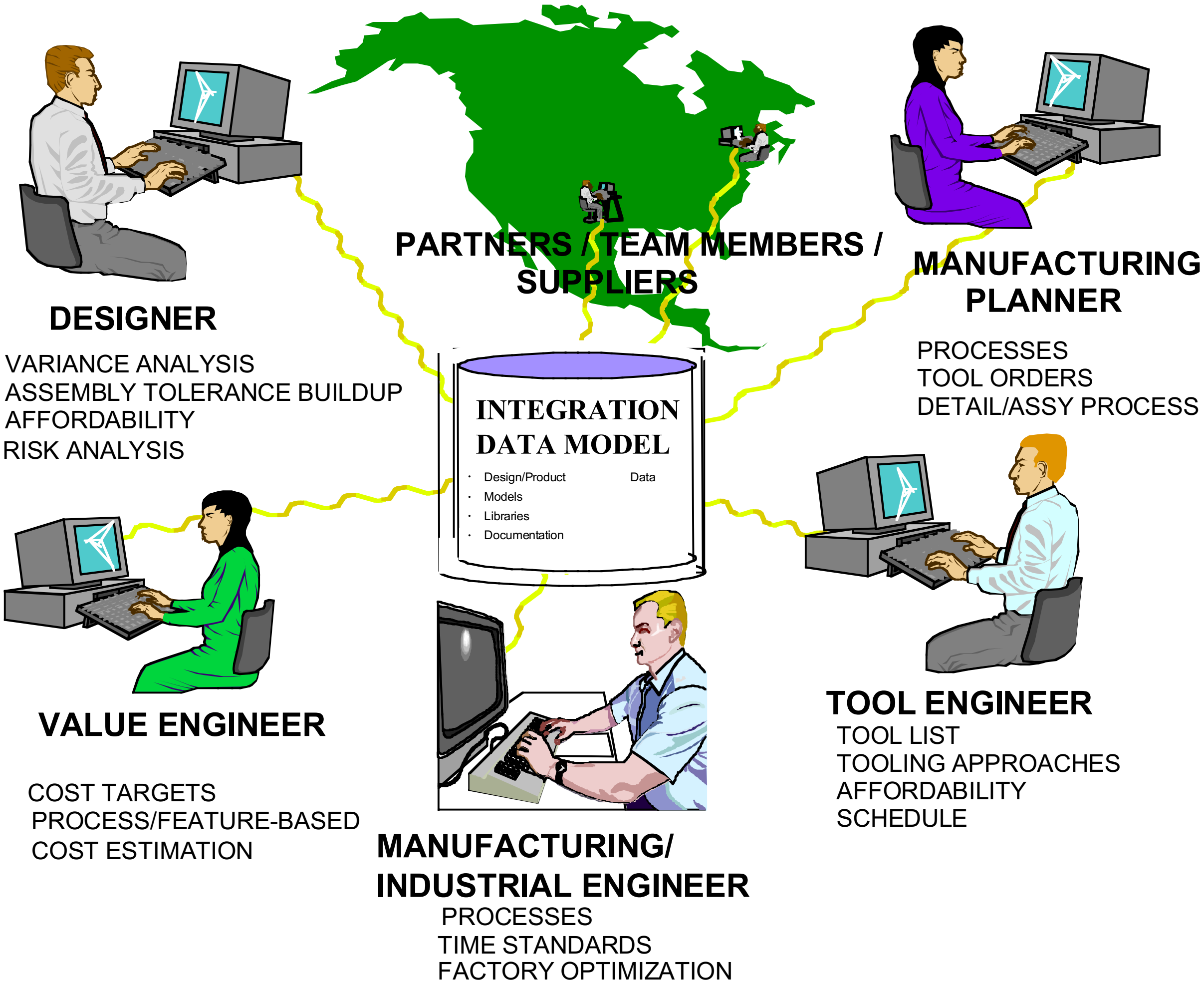


Figure B-1: IPT Use of Manufacturing Simulation Tools

Preconditions:

1. The organization has recognized the benefits of manufacturing simulation, and is committed to its use.

2. In large organizations, the design, systems engineering, manufacturing, and cost organizations recognize the significant advantages of sharing information to improve quality and timeliness of each function.

3. The design team includes members experienced in the use of the manufacturing simulation tools. Team members will, in some cases, operate more than one tool.

4. The design is done in a feature-based 3-D CAD tool. 3-D models are needed for visualization of simulations, and feature data is needed for tolerance analysis and costing.

Scenario:

Action	Software Reaction
Total Design Team meets to discuss design objectives and issues and to identify possible alternatives.	1. Mfg Team Lead creates and describes this study and possible alternatives in the integration database. 2. A new area in the Electronic Design Notebook (EDN) is setup for the team to share project management and technical data.
The Manufacturing Team may add additional process alternatives and establishes a simulation plan for each alternative. The team selects tools to be used, the order in which they will be executed (including iterations), and any options in each tool's input/output capability.	1. Additional alternatives are added to integration database. 2. Workflow information is described in workflow management tool. 3. Project schedules are created and noted in EDN.
The Manufacturing Team establishes naming conventions for operations, resources (tools and personnel), and any attributes that are used to extend the integration data model.	1. Naming conventions are documented in the EDN. 2. Mapping files that some tools use to translate shared information are updated.
If new tools are desired, they are acquired commercially or wrapped to be plug and play compatible with the integration data model.	1. New tools are installed and validated with test data.
Simulations, cost, and risk models are identified for use from previous studies or are developed or extended.	1. New simulation scripts and models are validated with test data.
The team checks that all required reference data (Reference process information, Materials data, Tool and Personnel resource information) are available and accessible from integration data model.	1. New reference data are loaded into the integration data model libraries.

Action	Software Reaction
Workflows associated with each design alternative are executed.	1. Workflow process is executed . 2. Workflow engine executes each task as defined, assuring that precedent tasks are complete before starting a task. 3. If an interactive tool is to be executed, the workflow engine sends an email to a cognizant user to inform them that the task is to be run. Batch tools are simply run. 4. Tool wrappers inform the workflow engine of their status. 5. Team members view the workflow status from their web browser.
Upon completion of the study of an alternative its status is changed from Working to In Review, and then Released.	1. The status of information in the integration data model is configuration managed and status flags are set by the Teal Lead to control read/write access to all data. 2. Simulation results are summarized in presentation format for review by team management. Summary information is added to EDN.
The Manufacturing Simulation and Overall Design Teams review alternatives and select the baseline design.	1. The selected baseline design alternative and selected alternative process plan are noted in integration data model. 2. Selection rationale is recorded in EDN.

Post Conditions:

Detailed information from the selected design alternative is accessible by downstream processes.

Simulation visualizations used for design are available for downstream use as animated work instructions, thus, reducing training time and errors.

Study data for alternatives not selected can be archived or deleted.

Exceptions:

None.

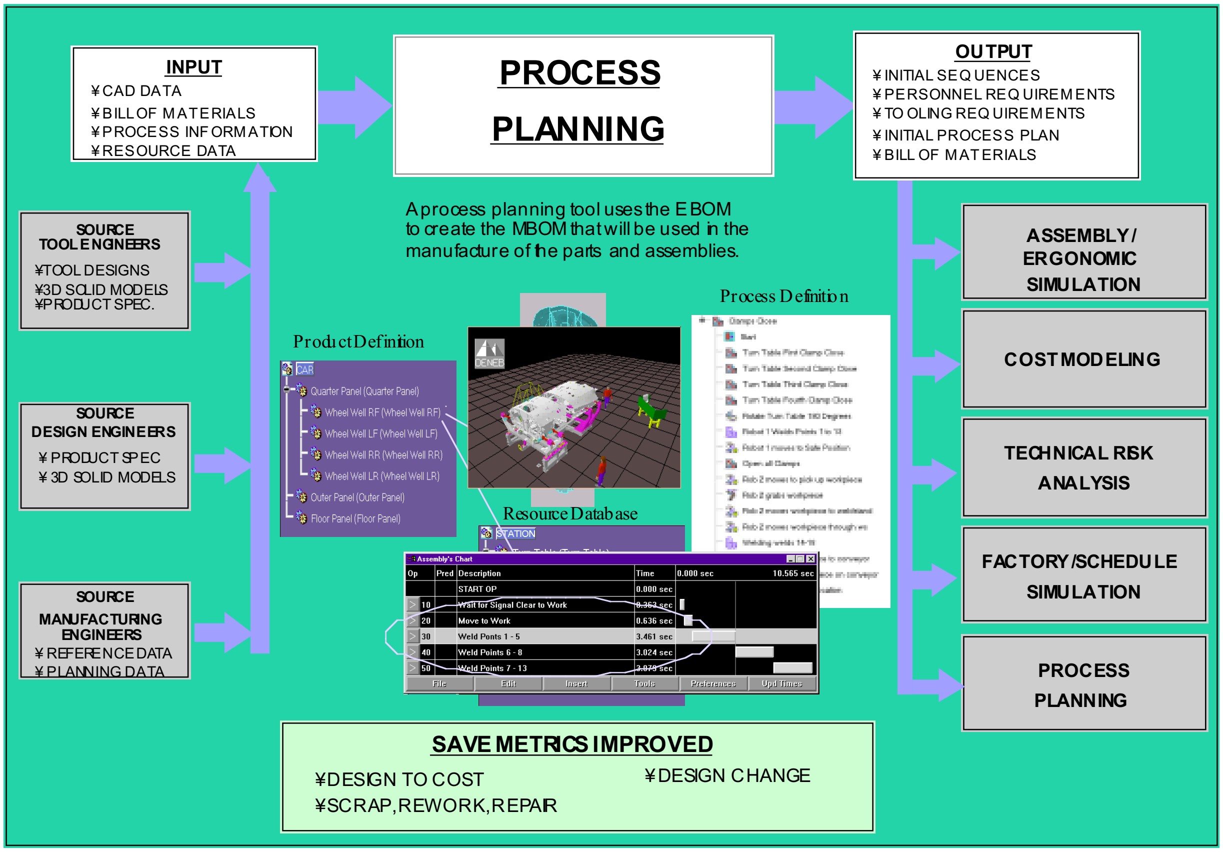
Relationship to other use cases:

Use cases for execution of each of the individual tool classes are included in this package.

Use Case: Process Planning

Overview:

This use case describes the interaction among a computerized Process Planning tool and various categories of manufacturing simulation and Computer Aided Design (CAD) tools. The Process Planning tool fills the very important role of creating the initial version of the manufacturing process and its individual operations and resources. This information is needed by nearly all of the other manufacturing simulation tools. The process planning tool takes the Engineering Bill of Material (EBOM) and available personnel and tool resources and creates the Manufacturing Bill of Material (MBOM) and process plan that will be used to manufacture the assembly. The process plan consists of a set of ordered operations or job steps, each of which define an elemental portion of the plan. In the planning process, the operations are defined, related to precedent operations, and assigned adequate resources (personnel and tools) to accomplish the task. This process plan, including all related information, is written to the Integration Data Model when it is complete. Many of the other simulation tools can then be run efficiently, without the need to manually re-enter the process plan data in different formats. As each tool refines some portion of the process plan, the design team may rerun the process planning tool to view and refine the plan, or to create new versions to study alternatives. The interaction of the process planning tool with other tools in the integrated manufacturing simulation system is illustrated in Figure B-2 below.



Op	Pred	Description	Time	0.000 sec 10.565 sec
		START OP	0.000 sec	
10		Wait for Signal Clear to Work	0.363 sec	
20		Move to Work	0.636 sec	
30		Weld Ponts 1 - 5	3.461 sec	
40		Weld Points 6 - 8	3.024 sec	
50		Weld Points 7 - 13	3.079 sec	

Figure B-2: Process Planning Tool Interactions

Preconditions:

1. Engineering Bill of Material for the assembly being studied is available as the input to the Manufacturing Bill of Material and process plan creation.

2. 3-D CAD models of the parts in an assembly are available to aid in the visualization of the sequence of the process plan.

3. Resources, both personnel and tools, that are available to the process are understood.

4. Reference data, such as standard process data, resources, material data have been populated in the integration data model libraries.

5. The design team agrees upon a workflow for the design study.

Scenario:

Action	Software Reaction
Designer creates a feature-based CAD model from which the Engineering Bill of Material is captured in electronic form.	EBOM is directly loaded in the integration data model or is accessible in or through the Product Data Management system.
Manufacturing Engineer reviews Engineering Bill of Material and decides on breakdown to major and subassemblies.	Process Planning tool accesses and displays EBOM in textual and graphical formats, and allows Mfg. Eng to change order of assembly.
Manufacturing Engineer identifies process steps that are required to accomplish each assembly step in the MBOM.	Information on standard or reference processes may be extracted from integration data model libraries.
Manufacturing Engineer identifies personnel and tool resources needed to accomplish each step of the process plan as it is related to Mfg Bill of Material.	Resources needed to accomplish plan are identified and related to manufacturing process plan steps.
MBOM, initial process plan and resources are saved.	Initial version of the process plan and all related information are written to integration data model to be used as basis for cost, schedule, and risk simulations.

Post Conditions:

The initial process plan has been loaded into the integration data model, ready to be used by other manufacturing simulation tools.

Common data is shared among CAD, Cost Modeling, Tolerance Analysis, Process Planning, Factory/Schedule Simulation and Assembly/Ergonomic Simulation tools.

Exceptions:

Recommendations may not be acceptable due to other constraints from either design or manufacturing, requiring further coordination.

Relationship to other use cases:

The Process Planning tool will typically be the first tool used to initiate a set of manufacturing simulations. It produces the initial version of the process plan, which is central to sharing data among the tools used to estimate cost, schedule, and risk of a proposed product or manufacturing process. Use cases for each of the other tools typically applied in a virtual manufacturing study are described in subsequent sections.

Use Case: Risk Analysis

Overview:

This use case describes the interaction among a Risk Analysis tool and various categories of manufacturing simulation and Computer Aided Design (CAD) tools. The risk analysis tool considered is distinctly different and much more rigorous than traditional systems engineering "High, Medium, and Low" risk methods. The risk tool allows the design team to create mathematical models of virtually any type of performance, cost, or schedule risk. Any input term in the risk model can be represented by a probability distribution that can be based on historical data or an expert's estimate. The risk tool rigorously propagates uncertainty to the desired final result through the model's set of equations. The design team then knows the probability of obtaining any given result within the range of possible outcomes. This approach is not meant to replace traditional Systems Engineering risk, but to supplement it when an accurately quantified risk is felt to be an important part of a design decision. Figure B-3 depicts the Risk Analysis tool with its inputs, outputs, and interactions within a portion of the product development environment.

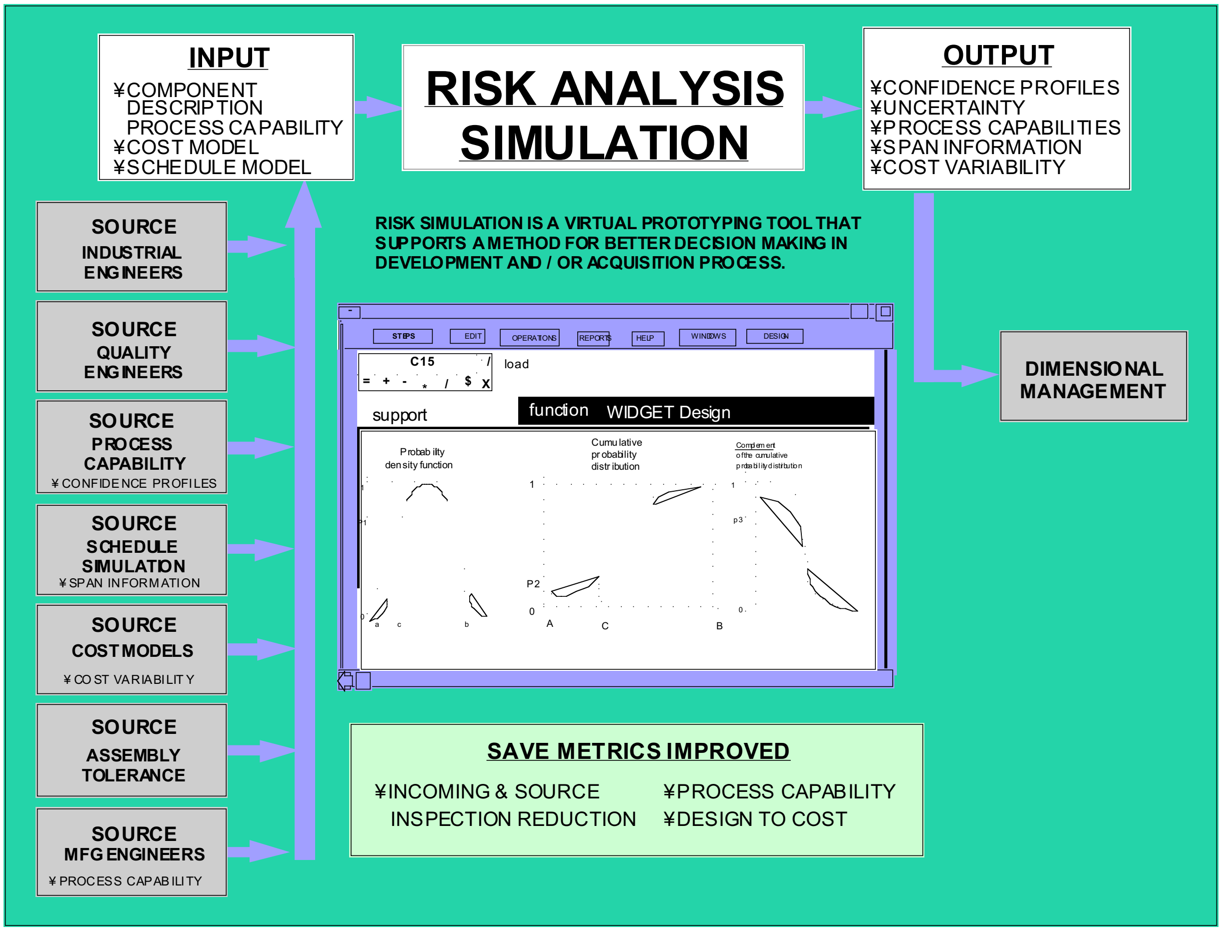


Figure B-3: Risk Analysis Tool Interactions

As the Risk Tool is simply a shell for development and execution of risk models, one or more team members will be responsible for developing the mathematical representation of the risk area(s) chosen for analysis. Equations represented in this model may come from any of the sources shown above. Much of the data fed into the model will come from the other tools integrated within the virtual manufacturing environment. Typically, the risk model will be structured around the manufacturing process plan and its set of individual operations. The inputs may be obtained from the integration data model that links the simulation tools. Depending on the risk area being assessed, these data can include reference process data or specific cost, schedule, or tolerance data from the case being simulated. The results from the risk analysis show process capabilities, confidence profiles, and uncertainty and are stored in the integration data model. These results can then be used by other simulations or as direct contributors to the final design decision process.

Preconditions:

1. The design team has identified an area of significant risk that requires more than a simple "high, medium, low" assessment.

2. A mathematical representation of the risk area is available or can be created.

3. Some portions of the risk model inputs are available from other manufacturing simulation tools.

Scenario:

Action	Software Reaction
Design team identifies areas of risk within design alternatives.	Execution of risk tool is planned and initial model equations created.
Process plan is generated manually or by computerized process planning tool.	Process plan is stored in integration data model.
Analyst develops risk model.	Process plan is read by risk tool and is used to automatically build structure of risk model.
Simulation tools are executed to add/refine data related to individual operations within process plan.	Simulation-based estimates of tolerance build-up, cost and schedule are saved in integration data model.
Risk is assessed by running model within risk tool.	1. Risk tool reads specific data required by risk model. 2. Model is executed . 3. Results are written to appropriate risk attributes within integration data model.
Action	**Software Reaction**
Risk data are used in subsequent simulations.	Other tools read risk data to refine estimates of cost or schedule.
Design team uses risk data to influence choice of baseline alternative.	Risk data are read from data model and recorded in electronic design notebook along with full rationale for selection of baseline alternative.

Post Conditions:

The team reviews results from all simulated design alternatives and a decision is reached.

Team lead sets data access status flags to block unwanted updates to information in integration data model.

Common data is shared among CAD, Cost Modeling, Tolerance Analysis, Process Planning, Factory/Schedule Simulation and Assembly/Ergonomic Simulation tools.

Exceptions:

Recommendations may not be acceptable due to other constraints from either design or manufacturing, requiring further coordination.

Relationship to other use cases:

Depending on the risk problem modeled, inputs to this use case can come from virtually any of the other tool use cases. Outputs from the risk model are used in the dimensional management model. This flexibility of the tools, and need to not constrain the design team in their use is a major challenge in the development of the integration data model.

Use Case: Cost Modeling

Overview:

This use case describes the interaction among a cost modeling tool and various categories of manufacturing simulation tools and Computer Aided Design (CAD) tools. A cost modeling tool provides a mechanism for building cost and producibility algorithms that evaluate a design based on its features, materials, resource requirements and manufacturing processes. The result of the analysis usually includes cost estimates, span times, product yields, and feedback on improvements in one or more areas of the design or process. Cost modeling, when used in conjunction with other design and analysis tools, can contribute to significant improvements in several key areas. Of course, employing cost tools throughout the product development process provides the opportunity to trade cost with other critical requirements to minimize cost impacts. This visibility into cost drivers for a particular product or design also helps to reduce design changes and rework while increasing the process capability. Figure B-4 depicts the cost simulation tool with its inputs, outputs, and interactions within a portion of the product development environment.

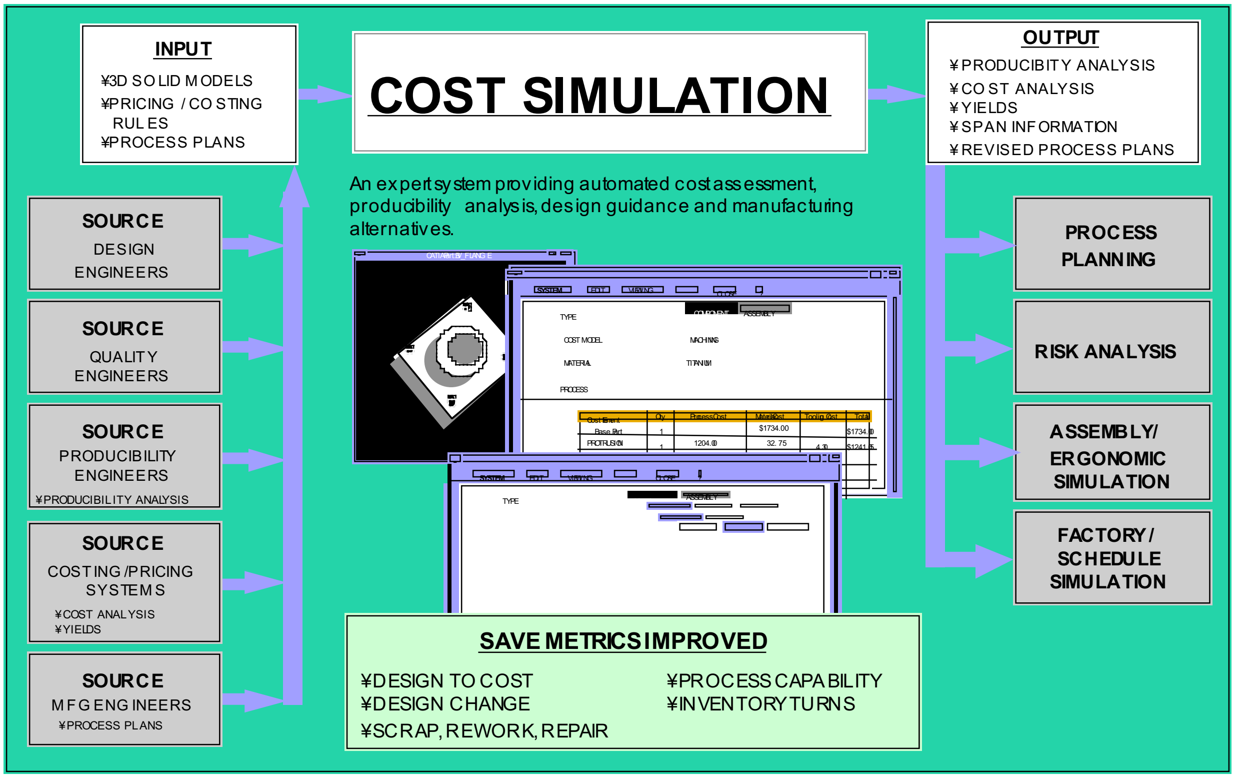


Figure B-4: Cost Modeling Tool Interactions

The flow of information is from the designer who creates the feature-based product geometry in a CAD system and the manufacturing engineer who creates the initial process planning data (e.g., manufacturing processes or assembly sequences) to the cost analyst who merges the design and the process using cost algorithms. The cost analyst uses these algorithms and the underlying knowledge bases to estimate the product and process cost. Other information, including process times and yields, are calculated during the cost analysis. The designer, the quality engineer, and the manufacturing engineer access these results in order to further analyze and modify the design and its manufacturing process.

Preconditions:

1. Designer uses a feature-based CAD package.

2. Cost estimation is not performed manually.

3. Development team uses manufacturing simulation tools.

4. Initial process planning is performed early in the development effort.

5. There is common or shared data among these tools.

Scenario:

Action	Software Reaction
Designer creates a feature-based CAD model.	Geometric definition is created with associated features.
Manufacturing engineer works with designer to define initial processes either in CAD system or external process planning tool.	Assembly sequences and/or manufacturing processes are created.
Cost analyst creates or accesses cost algorithms and knowledge bases.	Cost algorithms and knowledge bases are created in cost modeling tool format.
Cost analyst accesses CAD and process planning data.	Features and other geometric information are imported into cost modeling tool and process plans are extracted from the integration data model.
Cost analyst performs cost analysis.	Action internal to cost modeling tool. Cost algorithms and knowledge bases are used along with design and process information to calculate product and process costs.
Cost analyst makes cost, span, and yield information available.	Data is written to the integration data model.

Action	Software Reaction
IPT members access results and perform other simulations to refine data.	Process plan, cost, span and yield information is imported into simulation tools.
IPT uses cost data to influence the choice among alternatives.	Action external to software.

Post Conditions:

Cost drivers are identified when the cost to make design or process changes is minimal.

Common data is shared among Cost Modeling, CAD, Risk Analysis, Process Planning, Factory/Schedule Simulation and Assembly/Ergonomic Simulation tools.

Exceptions:

Recommendations may not be acceptable due to other constraints from either design or manufacturing, requiring further coordination.

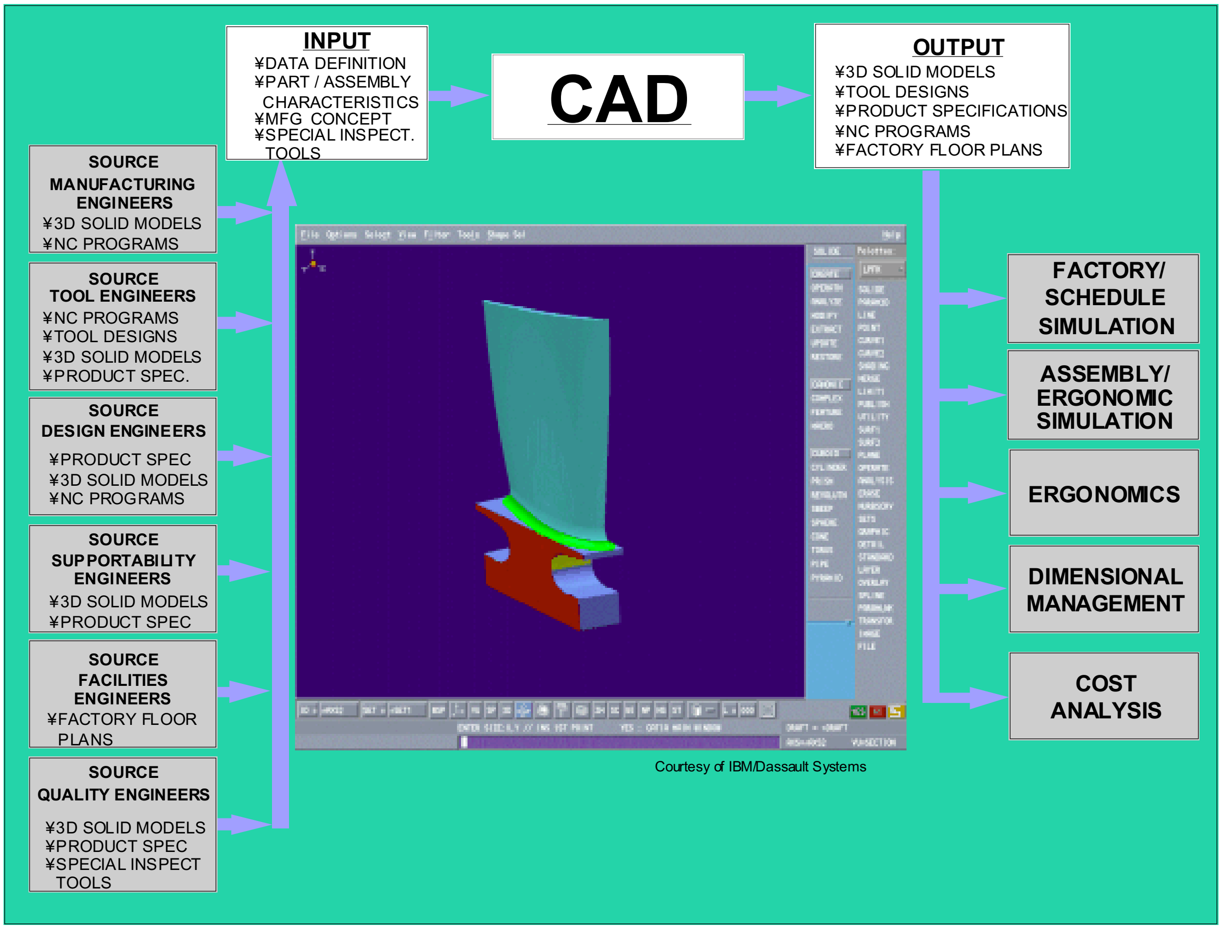
Relationship to other use cases:

The cost modeling tool may use inputs from any of the other tool use cases, especially where modifications to the design or process are identified. Outputs are essentially used for assessment of the alternatives being considered, but may be accessed by risk or factory/schedule simulations.

Use Case: Computer Aided Design / Manufacturing Simulation Facilitation

Overview:

This use case describes the interaction among a Computer Aided Design (CAD) tool and various categories of manufacturing simulation tools. CAD tools provide geometric information for part, assembly, tool, inspection and support equipment designs. This geometric information is produced from part and assembly definitions for a specific product in the form of three-dimensional, feature-based solid models. Factory/schedule simulation, assembly/ergonomic simulation, tolerance analysis, and cost modeling tools use the geometric information resulting from the CAD modeling activity to assess the design and recommend improvements. There are many ongoing efforts to ease the transfer of the geometric information itself to different types of analysis and simulation software; however, transfer of that information in the correct context for a given tool is not currently available. For example, both tolerance analysis and cost modeling tools use feature information in their assessment, but the types of features and the meaning of those features are vastly different in the context of each application. By making the CAD data available in the correct context, the drivers in specific areas of concern are more easily identified and addressed. Figure B-5 depicts the CAD tool with its inputs, outputs, and interactions within a portion of the product development environment.



Courtesy of IBM/Dassault Systems

Figure B-5: CAD Tool Interactions

The flow of information is from members of the design team, who provide product information, to the designer, who uses that information to create the feature-based product geometry in a CAD system. The manufacturing engineer uses this information, along with the appropriate tooling designs and factory layouts, to simulate the manufacturing processes, factory flow, or assembly sequences. In addition, the quality engineer uses the product definition to conduct tolerance and variation studies on the proposed design. The cost analyst uses the feature data from the CAD model as an input for estimating the product and process cost. Each of these users provides feedback to the designer about modifications that may improve the product design, thus, reducing the number of changes necessary later in the development process.

Preconditions:

1. Designer uses a feature-based CAD package.

2. Development team uses manufacturing simulation tools.

3. Early evaluation of product designs will yield improvements that would have been costly when identified later in the process.

4. There is common or shared data among these tools.

Scenario:

Action	Software Reaction
Designer creates a feature-based CAD model.	Geometric definition is created with associated features.
Manufacturing engineer accesses CAD model information for parts, tools, and factory layouts.	CAD models are brought into the factory/schedule simulation tool and the assembly/ergonomic simulation tool from the integration data model.
Manufacturing engineer performs simulations.	Action internal to simulation tools.
Manufacturing engineer makes recommendations for design or process changes.	New design or process information is created and saved in the integration data model.
Quality engineer accesses CAD model for part and tolerance data.	CAD models are brought into dimensional management tool.
Quality engineer performs tolerance analysis.	Action internal to dimensional mgt tool.
Quality engineer makes recommendations for design or process changes.	New design or process information is created and saved in the integration data model.
Action	**Software Reaction**
Cost analyst accesses CAD data.	Features and other geometric information are brought into cost modeling tool.
Cost analyst performs cost analysis.	Action internal to cost modeling tool.
Cost analyst makes recommendations for design or process changes.	New design or process information is created and saved in the integration data model.
Designer accesses design change recommendations.	Change information is brought into the CAD tool.
Designer evaluates changes with IPT members.	External to tools.
Designer incorporates changes.	Geometric information is updated in CAD tool.

Post Conditions:

Design changes are identified and incorporated with minimal impact to development process.

Common data is shared among CAD, Cost Modeling, Tolerance Analysis, Process Planning, Factory/Schedule Simulation and Assembly/Ergonomic Simulation tools.

Exceptions:

Recommendations may not be acceptable due to other constraints from either design or manufacturing, requiring further coordination.

Relationship to other use cases:

Most manufacturing simulation, especially those that are highly visual, use some form of the CAD data.

Use Case: Dimensional Management

Overview:

This use case describes the interaction among a Dimensional Management tool and various categories of manufacturing simulation tools and CAD Tools. The Dimensional Management software uses statistical simulation techniques to predict the amount of variation that can occur in an assembly due to specified design tolerances, fixture tolerances, and manufacturing/assembly variations. When tolerance assessments are conducted as part of the design evolution of components, assemblies, and tools, there is opportunity for significant improvements in product quality and cost. The Dimensional Management tool provides recommendations for optimal tolerances, assembly sequences, and support equipment requirements. Figure B-6 depicts the Dimensional Management tool with its inputs, outputs, and interactions within a portion of the product development environment.

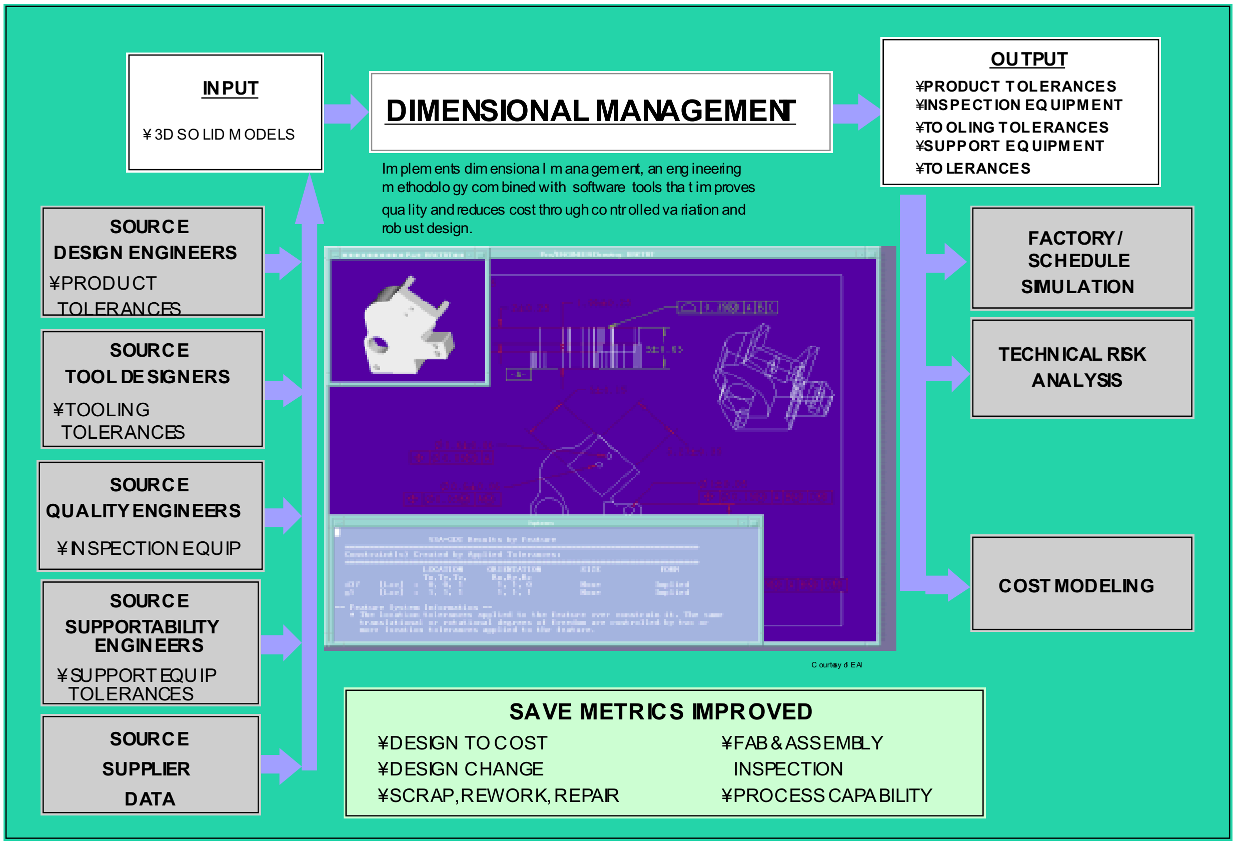


Figure B-6: Dimensional Management Tool Interactions

The flow of information is from the designer who creates the feature-based product and tooling geometry in a CAD system and the manufacturing engineer who works with the designer to define tolerance information for the components to the analyst who assesses the tolerance buildup and variation in parts and assemblies. The analyst uses the statistical simulation to determine the probability of achieving the desired part or assembly from a tolerance stack-up viewpoint. In addition the simulation provides a list of the major contributors to the variation.

The designer, the quality engineer, the cost analyst, and the manufacturing engineer access these results in order to further analyze and modify the design and its manufacturing process.

Preconditions:

1. Designer uses a feature-based CAD package and has defined tolerance information within the tool.

2. Initial process planning is performed early in the development effort.

3. Development team uses manufacturing simulation tools.

4. Early evaluation of product designs will yield improvements that would have been costly when identified later in the process.

5. There is common or shared data among these tools.

Scenario:

Action	Software Reaction
Designer creates a feature-based CAD model.	Geometric definition is created with associated features.
Designer works together with tolerance analyst to define tolerance data for parts and tools.	Tolerance data is added to geometric and feature definitions.
Manufacturing engineer works with designer to define initial processes either in CAD system or external process planning tool.	Assembly sequences and/or manufacturing processes are created and saved to the integration data model.
Tolerance analyst accesses the process plan.	Process plan is imported into dimensional management tool from the integration data model.
Tolerance analyst performs variational / tolerance analysis.	Action internal to dimensional management tool. Statistical methods are used to determine the probability of incurring problems due to variation.
Tolerance analyst interprets results from probabilities and contributor information.	Data is stored in the integration data model.

Action	Software Reaction
Tolerance analyst performs optimization to achieve acceptable results.	Action internal to dimensional management tool. Design specifications, tolerances, and process sequences are modified to achieve desired results.
Results are evaluated by cost, quality, manufacturing and design engineers.	Revised data imported into simulation tools and analyses performed. Assumption is that results are acceptable.
Design engineer incorporates changes.	Tolerance data and design changes are imported into CAD tool.
Manufacturing engineer incorporates changes.	Process plan modifications are imported into process planning tool from the integration data model.

Post Conditions:

Major variational contributors are identified when the cost to make design and process changes is minimal.

Common data is shared among Tolerance Analysis, CAD, Cost Modeling, Process Planning, Risk Analysis, and Factory/Schedule Simulation.

Exceptions:

Recommendations may not be acceptable due to other constraints from either design or manufacturing, requiring further coordination.

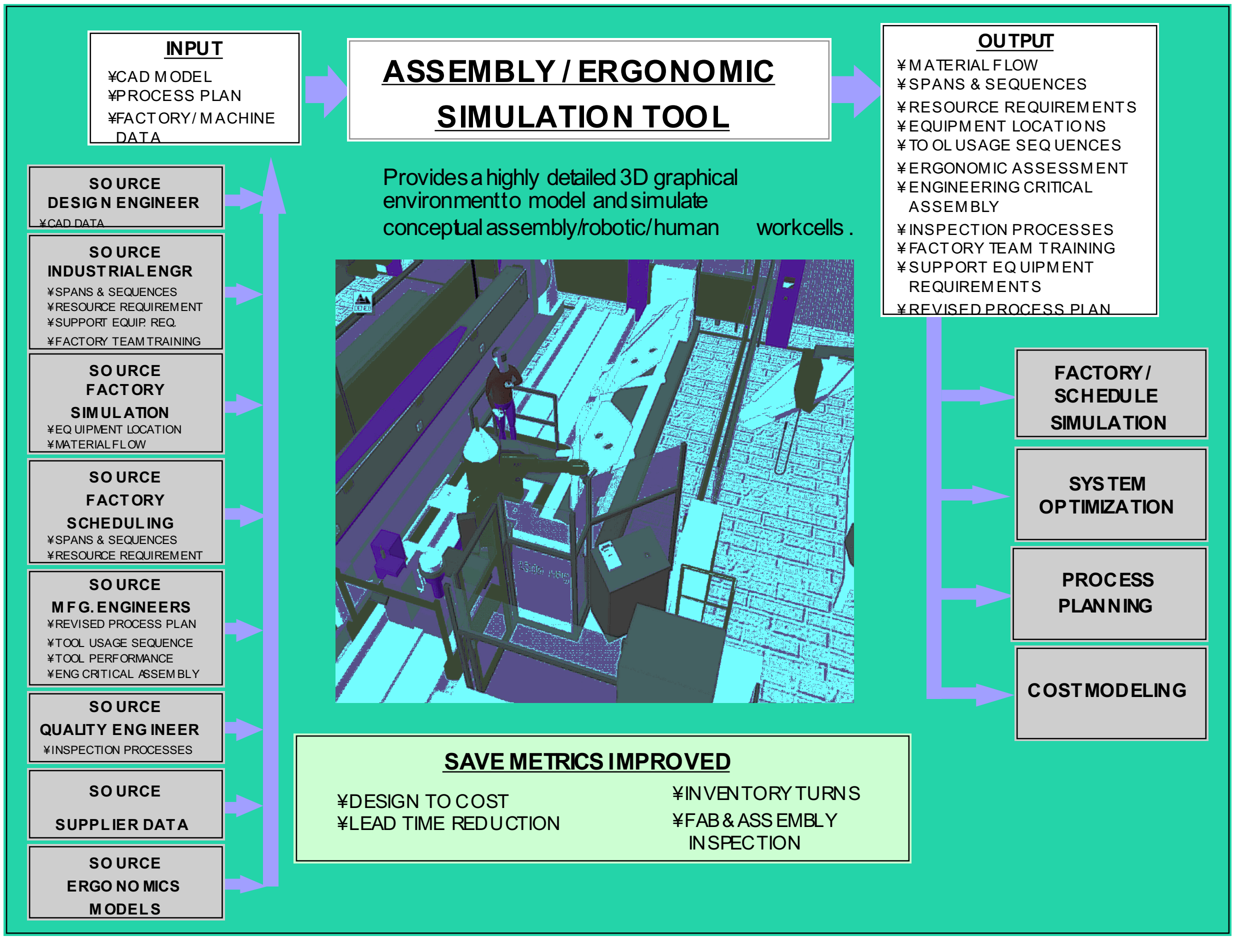
Relationship to other use cases:

CAD and process information is of key importance to creating a dimensional management simulation. Outputs may affect the design, process, and/or its capabilities and are evaluated by CAD, risk and factory simulation tools.

Use Case: Assembly/Ergonomic Simulation Tool

Overview:

This use case describes the interaction among an assembly/ergonomic simulation tool and various categories of manufacturing simulation tools and Computer Aided Design (CAD) tools. The Assembly/Ergonomic Simulation tool is used to simulate machinery, robots, and human interactions within an assembly work cell. The tool uses advanced three-dimensional graphics for visualization and provides an interactive environment in which to verify production concepts, work cell designs and manufacturing processes before implementing them on the shop floor. The resultant data usually includes optimum process sequences, resource requirements, ergonomic assessments and process times. Assembly/Ergonomic assessments, when used in conjunction with factory, cost, and risk simulations, can contribute to significant improvements in the assembly process, thus improving quality and avoiding downstream cost and schedule impacts. As an added benefit, the completed models may be used for simulation based training on the shop floor. Figure B-7 depicts the assembly/ergonomic simulation tool with its inputs, outputs, and interactions within a portion of the product development environment.



INPUT
¥CAD MODEL
¥PROCESS PLAN
¥FACTORY/ MACHINE DATA

ASSEMBLY / ERGONOMIC SIMULATION TOOL

Provides a highly detailed 3D graphical environment to model and simulate conceptual assembly/robotic/human workcells .

OUTPUT
¥MATERIAL FLOW
¥SPANS & SEQUENCES
¥RESOURCE REQUIREMENTS
¥EQUIPMENT LOCATIONS
¥TOOL USAGE SEQUENCES
¥ERGONOMIC ASSESSMENT
¥ENGINEERING CRITICAL ASSEMBLY
¥INSPECTION PROCESSES
¥FACTORY TEAM TRAINING
¥SUPPORT EQUIPMENT REQUIREMENTS
¥REVISED PROCESS PLAN

SOURCE
DESIGN ENGINEER
¥CAD DATA

SOURCE
INDUSTRIAL ENGR
¥SPANS & SEQUENCES
¥RESOURCE REQUIREMENT
¥SUPPORT EQUIP. REQ.
¥FACTORY TEAM TRAINING

SOURCE
FACTORY SIMULATION
¥EQUIPMENT LOCATION
¥MATERIAL FLOW

SOURCE
FACTORY SCHEDULING
¥SPANS & SEQUENCES
¥RESOURCE REQUIREMENT

SOURCE
MFG. ENGINEERS
¥REVISED PROCESS PLAN
¥TOOL USAGE SEQUENCE
¥TOOL PERFORMANCE
¥ENG CRITICAL ASSEMBLY

SOURCE
QUALITY ENGINEER
¥INSPECTION PROCESSES

SOURCE
SUPPLIER DATA

SOURCE
ERGONOMICS MODELS

FACTORY / SCHEDULE SIMULATION

SYSTEM OPTIMIZATION

PROCESS PLANNING

COST MODELING

SAVE METRICS IMPROVED
¥DESIGN TO COST
¥LEAD TIME REDUCTION
¥INVENTORY TURNS
¥FAB & ASSEMBLY INSPECTION

Figure B-7: Assembly/Ergonomic Simulation Tool Interactions

The flow of information is from the designer who creates the product geometry in a CAD system and the manufacturing engineer who creates the initial process planning data (e.g., manufacturing processes or assembly sequences) to the analyst who merges the design and the process into a three-dimensional simulation environment. The analyst creates the motion attributes, kinematics and input/output logic to produce the simulations. Along with the visualization results for assembly and ergonomic issues, process times are calculated for the modeled sequence. The designer, the quality engineer, the cost analyst and the manufacturing engineer access these results in order to further analyze and modify the design and its manufacturing process.

Preconditions:

1. Designer uses a CAD package.
2. Initial process planning is performed early in the development effort.
3. Development team uses manufacturing simulation tools.
4. There is common or shared data among these tools.

Scenario:

Action	Software Reaction
Designer creates a CAD model.	Geometric definition is created.
Manufacturing engineer works with designer to define initial processes either in CAD system or external process planning tool.	Assembly sequences and/or manufacturing processes are created and stored in the integration data model.
Manufacturing engineer accesses CAD data.	Geometric information is imported into cost assembly/ergonomic simulation tool.
Manufacturing engineer accesses process planning data.	Process plan is imported into assembly/ergonomic simulation tool from the integration data model.
Manufacturing engineer builds simulation.	Action internal to assembly/ergonomic simulation tool. Geometric and sequencing information are merged with personnel and tooling requirements to create the simulation.
Manufacturing engineer runs simulation.	Action internal to assembly/ergonomic simulation tool. Assumption is that results are not acceptable.
Action	**Software Reaction**
Manufacturing engineer makes recommendations for design or process changes.	New design or process information is created and saved.
Recommendations are reviewed by cost, quality, and design engineers.	Revised data imported into simulation tools and analyses performed. Assumption is that results are acceptable.
Manufacturing engineer accesses recommendations.	New process imported into process planning tool from the integration data model.

Post Conditions:

Assembly "optimization" is performed when the cost to make design or process changes is minimal.

Possible ergonomic issues are identified and addressed prior to sending to the shop floor.

Training material is available to shop floor personnel.

Common data is shared among Assembly/Ergonomic Simulation tools, CAD, Cost Modeling, Risk Analysis, Process Planning, and Factory/Schedule Simulation.

Exceptions:

Recommendations may not be acceptable due to other constraints from either design or manufacturing, requiring further coordination.

Relationship to other use cases:

Assembly/Ergonomic simulation relies heavily on information from the CAD and process planning use cases to create the simulation. Resulting process times are used for cost, risk, and schedule estimates. Revised process steps are used by CAD and process planning.

Use Case: Factory/Schedule Simulation Tool

Overview:

This use case describes the interaction among a factory/schedule simulation tool and various categories of manufacturing simulation tools and Computer Aided Design (CAD) tools. The Factory/Schedule Simulation tool is used to simulate real-world system behaviors including factory flow, capacity planning, and production scheduling. The tool uses graphics for visualization and provides an interactive environment in which to assess the productivity, cost-effectiveness, and efficiency of proposed manufacturing systems. The resultant data includes production rates, status of factory components, schedules, and alternative factory layouts. The factory/schedule simulations, when used in conjunction with process planning, cost, and risk simulations, can contribute to significant improvements in process flow, thus reducing cost and scheduling impacts. Figure B-8 depicts the factory/schedule simulation tool with its inputs, outputs, and interactions within a portion of the product development environment.

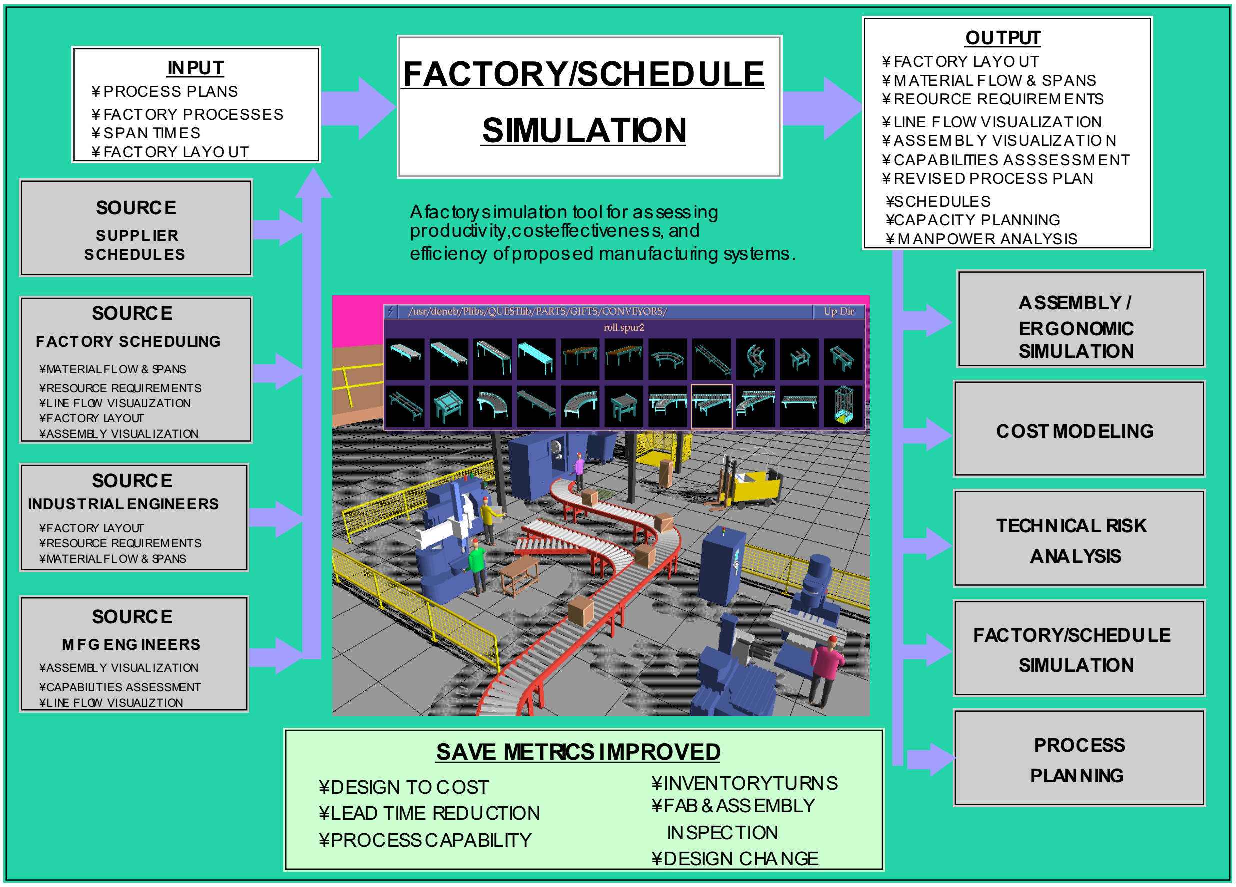


Figure B-8: Factory/Schedule Simulation Tool Interactions

The flow of information is from the designer who creates the product geometry in a CAD system and the manufacturing engineer who creates the initial process planning data (e.g., manufacturing processes or assembly sequences) to the analyst who merges the design and the process, along with the factory layout, into a simulation environment. The analyst creates factory flow activities, including routing, sequencing, and merging, to complete the simulation model. Along with the visualization results for factory flow, schedules, resource information, and revised process flows are calculated for the modeled sequences. The designer, the quality engineer, the cost analyst and the manufacturing engineer access these results in order to further analyze and modify the design and its manufacturing process.

Preconditions:

1. Designer uses a CAD package.

2. Initial factory process planning is performed early in the development effort.

3. Development team uses manufacturing simulation tools.

4. There is common or shared data among these tools.

Scenario:

Action	Software Reaction
Designer creates a CAD model.	Geometric definition is created.
Manufacturing engineer works with designer to define initial processes either in CAD system or external process planning tool.	Assembly sequences and/or manufacturing processes are created and stored in the integration data model.
Manufacturing engineer accesses CAD data.	Geometric information is imported into factory/schedule simulation tool.
Manufacturing engineer accesses process planning data.	Process plan is imported into factory/schedule simulation tool from the integration data model.
Manufacturing engineer builds simulation.	Action internal to factory/schedule simulation tool. The geometric and process planning information is combined with defined path and resource requirements.
Manufacturing engineer runs simulation and makes recommendations for design or process changes.	New process plan is saved to integration data model.

Action	Software Reaction
Recommendations are reviewed by cost, quality, and design engineers.	Revised data imported into simulation tools and analyses performed. Assumption is that results are acceptable.
Manufacturing engineer performs new simulations and makes schedule data available.	Resultant schedule data is stored in the integration data model.

Post Conditions:

Factory flow "optimization" is performed when the cost to make design or process changes is minimal.

Common data is shared among Factory/Schedule Simulation, CAD, Cost Modeling, Risk Analysis, Process Planning, and Assembly/Ergonomic Simulation tools.

Exceptions:

Recommendations may not be acceptable due to other constraints from either design or manufacturing, requiring further coordination.

Relationship to other use cases:

Primary inputs to the factory/schedule simulation are the CAD data and process plan with associated resource and tooling requirements. Schedule outputs are used by the IPT to make an overall assessment of the alternative designs and processes.

Appendix C

SAVE Data Model Dictionary

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

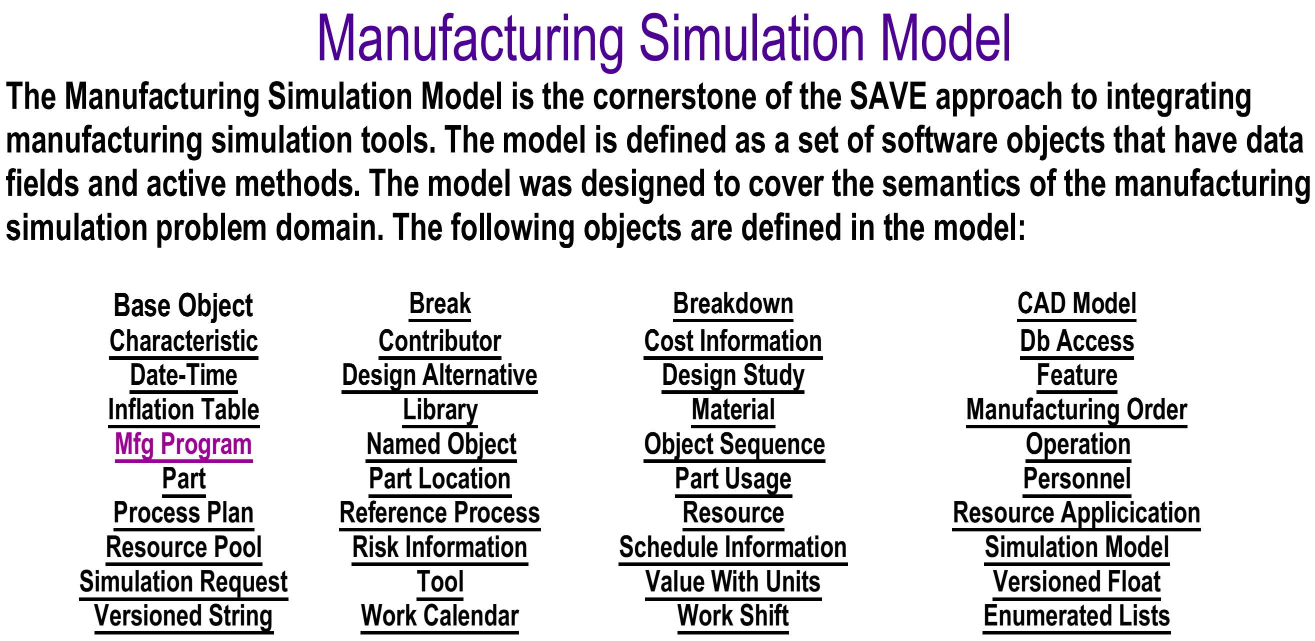
1.0 Introduction

The SAVE Data Model (SDM) is the key integration element of the SAVE system. The manufacturing simulation tools communicate with each other by reading and writing to this model as it is implemented in the SDM server software. The SDM is formally specified in the CORBA Interface Definition Language (IDL). The SDM dictionary included below is a non-technical version of the IDL meant for end users to learn the capability that is provided. It is important for users to understand the model to help them decide in which order to execute their simulation tools to maximize data sharing.

The pages shown below are images of a set of web pages that are available on the SAVE website (http://skipper.mar.external.lmco.com/save). While they can be reviewed in printed form, the interactive pages have the advantage that references from one data object to another are hyperlinked and it is easy to move around the model. Users have found that the interactive version of the dictionary operates in a way that is very analogous to the actual SAVE object-oriented data model. Users with flat file or relational database experience came to understand the object-oriented approach quickly by using the web pages.

2.0 Textual Index

Figure C-1 shows the textual index to all of the data objects in the SDM. On the web pages, each of these names is linked to the complete definition of that data object.



Manufacturing Simulation Model

The Manufacturing Simulation Model is the cornerstone of the SAVE approach to integrating manufacturing simulation tools. The model is defined as a set of software objects that have data fields and active methods. The model was designed to cover the semantics of the manufacturing simulation problem domain. The following objects are defined in the model:

Base Object	Break	Breakdown	CAD Model
Characteristic	Contributor	Cost Information	Db Access
Date-Time	Design Alternative	Design Study	Feature
Inflation Table	Library	Material	Manufacturing Order
Mfg Program	Named Object	Object Sequence	Operation
Part	Part Location	Part Usage	Personnel
Process Plan	Reference Process	Resource	Resource Applicication
Resource Pool	Risk Information	Schedule Information	Simulation Model
Simulation Request	Tool	Value With Units	Versioned Float
Versioned String	Work Calendar	Work Shift	Enumerated Lists

Figure C-1: Textual Index to SAVE Data Model

The SDM Dictionary also includes a graphical index, Figure C-2, that first provides a high-level overview of the model and is again hyperlinked to the individual data objects. In the graphical index, the key user objects are shown in context, and all data objects can be accessed by traversing from the user objects to the lower-level supporting objects.

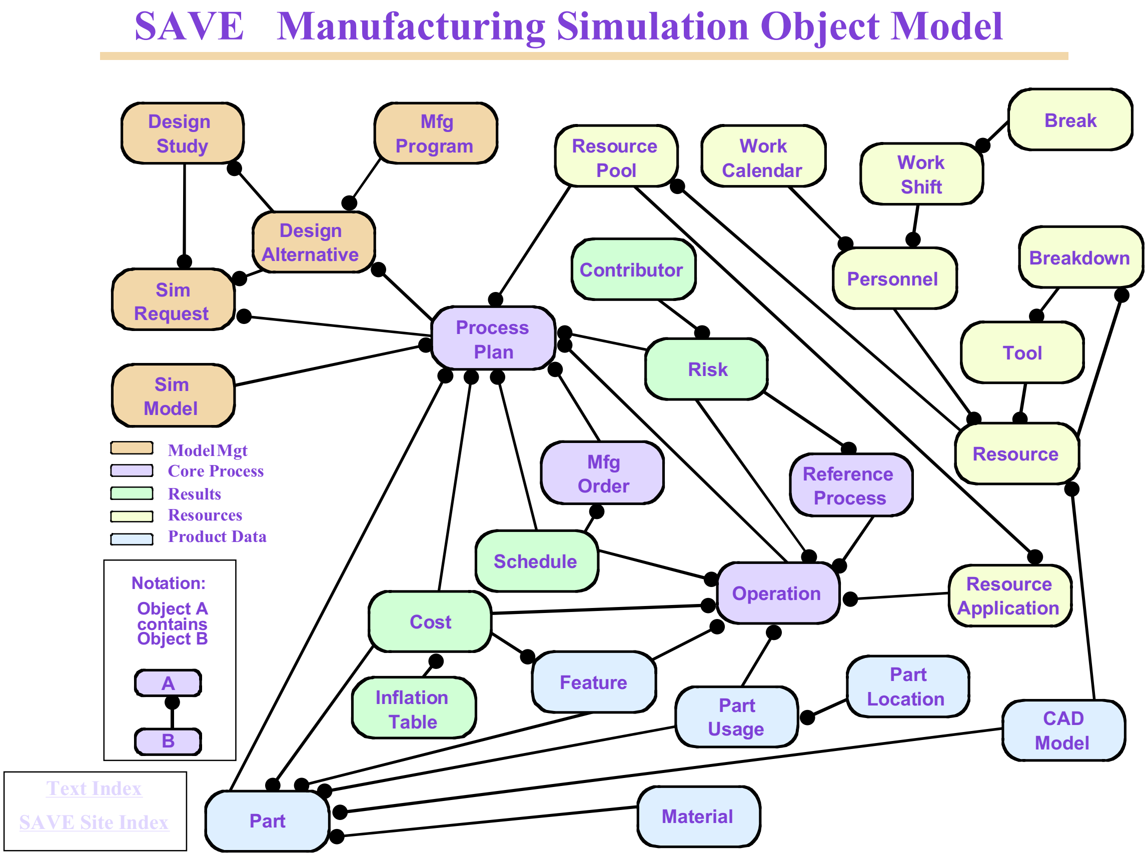


Figure C-2: Graphical Index to SAVE Data Model

Each of the SDM data objects is described below.

BASE OBJECT

msmBaseObject

This is the base object of all Manufacturing Simulation Model (msm) objects. It allows definition of data fields and methods that are common to all objects. Users need not be concerned with this object, as all user data fields are shown in each object in this set of linked pages.

Data Fields

Name	Data Type	Description
ObjType	ObjectType	Each object knows its type.
SetRef	Text	Used by server to support persistence.

Methods

None Defined for Base Object

Inherits From: None
Used In: All msm objects

BREAK

MsmBreak

This interface defines the start and end times associated with a Work Shift break..

Data Fields

Name	Data Type	Description
Name	String	User defined name for this break.
Description	String	User defined description for this break.
DateTime	DateTime	Date and time when this break object was created.
StartTime	Real Number	Break start time in hours after midnight.
EndTime	Real Number	Break end time in hours after midnight.

Methods

None Defined for Break

Inherits From: NamedObject
Used In: Work Shift

BREAKDOWN

Breakdown

This interface defines the breakdown (in terms of failure to operate) characteristics of a tool. It also contains the resource required to repair the breakdown. The name and description attributes contain the mode, or manner of failure of the breakdown.

Data Fields

Name	Data Type	Description
Name	Text	Name of the breakdown.
Description	Text	Textual description of the breakdown. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a breakdown is created.
HrToFirstFail	Real Number	Mean time to first failure in hours.
MHrBF	Real Number	Indication of the amount of time between breakdowns for a given tool (mean time between failure) in hours.
RepairHr	Real Number	Time to repair in hours.
RepairResource	Sequence of Resource	List of resources needed to repair tool. Resources may be Tools or Personnel.

Methods

None Defined for Breakdown.

Inherits From: Named Object
Used In: Tool

CAD MODEL

msmCADModel

This object contains descriptive and location information about the graphical representation of models used for simulations (includes parts, tools, personnel, etc.). The database maintains a list of all CAD models in one of its Libraries. Each new CAD Model is automatically placed in the Library when it is created.

Data Fields

Name	Data Type	Description
Name	Text	Name of the CAD model.
Description	Text	Textual description of the CAD model. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a CAD model is created.
EnvelopX	Real Number	X dimension (height) of the envelope that would fully enclose the model - Inches.
EnvelopY	Real Number	Y dimension (height) of the envelope that would fully enclose the model - Inches.
EnvelopZ	Real Number	Z dimension (height) of the envelope that would fully enclose the model - Inches.
Format	Text	Storage format for the CAD model. Could be native CAD for standards-based (ex STEP)
Location	Text	Storage location for the CAD model. Computer node and path information.

Methods

None Defined for CAD Model

Inherits From: Named Object
Used In: Part, Resource, Simulation Model

CHARACTERISTIC

msmCharacteristic

Characteristics contains a name / value / units that is used in many other objects to flexibly expand the list of data fields. The Numeric Value **object associated with a Characteristic will be automatically created when the Characteristic is created, and only needs to be populated if needed.**

Data Fields

Name	Data Type	Description
Name	**Text**	**Name of the characteristic.**
Description	**Text**	**Textual description of the characteristic. Content is at user's discretion.**
DateTime	Date Time	**Date and time assigned by the system when a characteristic is created.**
TextValue	**Text**	**Textual value of a Characteristic.**
NumericValue	Value With Units	**Numerical value of a Characteristic. This object knows its units and can perform conversions if necessary.**

Methods

None Defined for Characteristic.

Inherits From: Named Object
Used In: Process Plan, Operation, Reference Process, Tool, Risk, Material, Feature

CONTRIBUTOR

msmContributor

This object defines a contributor and its percentage contribution to a RiskInfo object. PercentContribution will be set to 0 when the contributor is created.

Data Fields

Name	Data Type	Description
Name	Text	Name of the contributor.
Description	Text	Textual description of the contributor. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a contributor is created.
PercentContribution	Versioned Real Number	Percentage of the total risk that is due to this contributor.

Methods

None Defined for Contributor.

Inherits From: Named Object

Used In: Risk Info

COST INFORMATION

msmCostinfo

This object contains the basic set of cost data used throughout the model. All values are in US dollars consistent with the "FiscalYear" value. All variables are stored as VersionedFloat types to allow versioning of the data. The cost interface is used at various levels of the model.

Data Fields

Name	Data Type	Description
Type	Costtype	Type of cost being stored, i.e. First Unit, Average, or Total. This data field is defaulted to Average.
FiscalYear	Integer	Year for which the costs are estimated.
BaseYear	Integer	Year for which inflation values are 1.0. This year must the current year, or earlier.
MaterialInflationFactor	Real Number	Factor used to inflate the material cost from the base year to the fiscal year.
LaborInflationFactor	Real Number	Factor used to inflate the labor cost from the base year to the fiscal year.
InflationTable	InflationTable	Table of inflation values used to adjust costs to other years.
RecurringMfgLabor	Versioned Float	The repetitive manufacturing labor costs expended in the fabrication, assembly, and test of a hardware product. It includes field operations, mod & test support.
OtherRecurringMfg	Versioned Float	Recurring cost associated with such elements as travel, photographic services, facility rental, multi-media, etc.
OtherNonRecurringMfg	Versioned Float	Non-recurring cost associated with elements such as travel, in country program office, overtime premium, reproduction, etc.
MaterialCost	Versioned Float	Cost for raw materials, purchased parts, and standard hardware.
QualityAssureance	Versioned Float	Cost associated with activities such as process inspection, nondestructive inspection, records keeping/maintenance, quality action reporting.
DevelopmentTooling	Versioned Float	Cost associated with the tooling required during a product's development phase. The tooling concepts used generally are incomplete and/or low production rate quality tools.
ProductionTooling	Versioned	The cost of tooling that is generally made of the

	Float	best and most practical materials available. They are capable of producing parts with critical tolerances at an accelerated production rate with relatively little to no additions or changes in design.
NonrecurringTooling	Versioned Float	Cost of fixtures, molds, jigs, electronic media, and mfg. Aids acquired or manufactured by a contractor.
ToolMaterial	Versioned Float	Material cost associated with the raw material required to construct the tool in-house or the supplier's material charge to construct the tool.
SustainingTooling	Versioned Float	Cost associated with engineering and manufacturing tool maintenance. Some of the activities within the engineering area are tool design, methods planning, and liaison. Some of the activities within the manufacturing area are repair, refurbish and incorporation of changes to existing tools, and tool storage.
PlantEquipment	Versioned Float	Cost associated with capital equipment such as autoclaves, portable tools, vehicles, and machinery that are used in the manufacture, research and development, and/or test of products or general plant purpose.
TotalMfgCost	Versioned Float	The cost of the mfg labor and materials required to build the product end item. This cost is generally a basic functional cost category that does not include tooling cost.
AvgProductionCost	Versioned Float	This cost is the arithmetic mean of the number of units to be built and represents a cost of a unit that has not incurred any start up problems typically associated with a production run.

Methods

None Defined for Cost Info.

Inherits From: Base Object
Used In: Process Plan, Operation, Part, Feature

DATE TIME

msmDateTime

This object implements a standard date-time format for this model.

Data Fields

Name	Data Type	Description
DateTime	Text	This read-only data field returns the date-time as a string of text in the format yyyy/mm/dd 24:mm:ss.

Methods

Name	Returned Value Type	Parameters (I/O/IO)Name/Type	Method Description
SetCurrentDateTime	None	None	Sets the stored date-time to the current value of the system clock.
SetDateTime	None	(I) / DT / Text	Sets date time to the value specified in the input text string. Valid format is yyyy/mm/dd 24:mm:ss.
ValidateDateTime	True/False	(I) / DT /Text	Validates a date-time text string. Valid format is yyyy/mm/dd 24:mm:ss.
ParseDateTime	None	(I) / DT / Text Outputs (all Integers): Year, Month, Day, Hour, Minute, Sec	Inputs a Date-Time text string and returns the numeric values.
CreateDateTime	Text	Year, Month, Day, Hour, Minute, Sec - All are Integer inputs	Creates a date-time string from numeric inputs.

Inherits From: Base Object
Used In: All Named Objects

DATABASE ACCESS

msmDbAccess

This object provides the operations that control database access including Read and Update transaction control, Commit or Rollback, and a general object locate capability. This interface contains no data attributes and is not made persistent in the data store. The client in a client-server system will create and use a Database Access object to control database activity.

Data Fields

None defined for Database Access.

Methods

Name	Returned Value Data Type	Parameters (I/O/IO)/Name/Data Type	Method Description
Opendatabase	None	None	Initiates database access for client.
BeginReadTrans	None	None	Initiates a logically related set of data read operations. Other read and (one) write operation can run in parallel with this access.
EndReadTrans	None	None	Ends the read transaction.
BeginUpdateTrans	None	None	Initiates a database update transaction (set of read and/or write operations). This operation locks out other write operations, but allows other reads.
Commit	None	None	Commits the results of the update transaction to the database.
Rollback	None	None	Rolls back the current update transaction. Changes made since the BeginUpdateTrans are not made permanent in the database.
GetLibrary	Library	(I)/Lib/Text	Returns a library object of a desired type.
GetSimReqst	Simulation Request	(I)/Nam/Text	Returns a Simulation Request object to a client. This object contains information to tell a simulation code where in the database to get/put its simulation data.

Distribution Statement A: Approved for public release. Distribution is unlimited.

CreateObject	BaseObject	(I)/Typ / ObjectType (I) Nam / String (I) Desc / String	This method is used to create NamedObjects of any type.
CopyObject	BaseObject	(I)/ refObject / BaseObject (I) Nam / String (I) Desc / String	Copy an object into a new object. Only the following objects can be copied: Break, Breakdown, DesignAlternative, DesignStudy, InflationTable, Material, MfgProgram, Part, Personnel, ProcessPlan, RefProcess, SimReqst, Tool, WorkCalendar, WorkShift

Inherits From: Does not inherit from another object.
Used In: Used by client software, not other objects.

DESIGN ALTERNATIVE

msmDesignAlternative

A design study alternative defines one approach to a design or process decision. Typically a design study will include a small number or alternatives, but this model places no limit on the number of alternatives. Constructor will create ProcessPlans sequence. User must add MfgProg after finding one in the library or creating one in the factory.

Data Fields

Name	Data Type	Description
Name	Text	Name of the design alternative.
Description	Text	Textual description of the design alternative. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a design alternative is created.
DAStatus	Status	Flag set to handle high-level configuration management. Allowable values are working, review, and released.
BaselineProcPlan	Process Plan	Alternative deemed to be the most desirable in terms of the criteria defined for the study. For example, this could be cost, schedule, and risk related.
MfgProgram	Manufacturing Program	Associated manufacturing program information.
ProcessPlans	Sequence of Process Plan	List of alternative Process Plans associated with this design alternative.

Methods

None Defined for Design Alternative

Inherits From: Named Object

Used In: **Design Study**, **Simulation Request**

DESIGN STUDY

msmDesignStudy

A design study provides a container for data associated with a manufacturing design study. It includes the evaluation of one or more alternatives with respect to specific variables. Design studies are the top-level data object within this model and provide overall configuration management controlled by the "Status" attribute.

Data Fields

Name	Data Type	Description
Name	Text	Name of the design study.
Description	Text	Textual description of the design study. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a design study is created.
DSStatus	Status	Flag set to handle high-level configuration management. Allowable values are working, review, and released.
Selected Alternative	Design Alternative	Alternative deemed to be the most desirable in terms of the criteria defined for the study. For example, this could be cost, schedule, and risk related.
Alternative	Sequence of Design Alternatives	Set of 1 or more design study alternatives associated with a particular design study.
Summary URL	Text	URL (location) of the web page that contains summary information for the design study.

Methods

None Defined for Design Study

Inherits From: Named Object
Used In: Simulation Request

Manufacturing Simulation Model

Enumerated Lists Used In Model

Name	Description	Valid Values
Status	Object configuration control flag	Working, Review, Released
PartType	Declares the type of a part	Detail, Assembly
DaysOfWeek	Days of the week	Sun, Mon, Tue, Wed, Thu, Fri, Sat, Sun
Months	Months of the year	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec
SimType	Simulation code types	Enterprise, CAD, ProcessPlanning, Schedule, Assembly, Factory, Cost, Schedule, Risk
CostType	Types for cost info	FirstUnit, Average, Total
TextQual	Textual qualitative rating	Low, Med, High
SeqType	Type of object sequences	DupNames, UniqueNames
ObjectType	Objects supported Mfg Sim Model	BaseObject, Break, Breakdown, CADModel, Characteristic, Contributor, CostInfo, DbAccess, DateTime, DesignAlternative, DesignStudy, Feature, InflationTable, Library, Material, MfgOrder, MfgProgram, NamedObject, ObjectSeq, Operation, Part, PartLocation, PartUsage, Personnel, ProcessPlan, RefProcess, Resource, ResourcePool, ResourceApplic, RiskInfo, SchedInfo, SimModel, SimReqst, Tool, ValueWithUnits, VersionedFloat, VersionedString, WorkCalendar, WorkShift

FEATURE

msmFeature

This object defines a set of data fields that are used by all part features. These features can be either design or manufacturing features depending on their use. Cost and Sequence of Characteristics objects are automatically created when a Feature is created, but remain empty until client software populate them.

Data Fields

Name	Data Type	Description
Name	Text	Name of the feature.
Description	Text	Textual description of the feature. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a feature is created.
Quantity	Integer	Quantity of this feature in a given part.
Cost	Cost Info	Cost of this feature. This is the total cost for all instances of this feature if Quanty is greater than 1.
Type	Text	Feature type - may be a base part or other specific feature such as a hole.
Characteristics	Sequence of Characteristics	List of Name-Value pairs that allow users to flexibly expand the information associated with a Feature.

Methods

None Defined for Feature.

Inherits From: Named Object
Used In: Tool, Personnel

INFLATION TABLE

msmInflationTable

This object contains a table of inflation values for a range of years. This is used to define the inflation assumed in forward year dollars. Inflation tables are maintained in the Manufacturing Simulation Model Library. Newly created tables are automatically placed in this library.

Data Fields

Name	Data Type	Description
Name	Text	Name of the inflation table.
Description	Text	Textual description of the inflation table. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a inflation table is created.

Methods

Name	Returned Data Type	Parameters (I/O/IO) / Name / Type	Method Description
AddYear	None	(I)/ Year / Integer (I) / Rate / Real No.	Adds a year and its inflation value to the list.
ChangeYear	None	(I)/ Year / Integer (I) / Rate / Real No.	Changes data for a year.
GetRate	Real Number	(I) / Year / Integer	Returns the inflation rate for a given year.
SetExtrapRate	None	(I) / Extrap / Real No	Sets inflation rate for use beyond last year in table.
InflateCost	Real No	(I)/ Orig Year / Integer (I) / Cost / Real No. (I) / New Year / Integer	Returns the cost adjusted to the desired year.

Inherits From: Named Object
Used In: Cost Info

LIBRARY

msmLibrary

This object defines a list with access methods that is used to store references to all instances a particular type of object. The DBAccess object has a method to locate a particular Library object. Library objects are created when a database is initialized, although new Libraries can be created at any time.

The following Libraries are currently defined:

- Break
- CAD Model
- Design Alternative
- Design Study
- Inflation Table
- Material
- Manufacturing Program
- Personnel
- Part
- Process Plan
- Reference Process
- Simulation Request
- Tool
- Work Calendar
- Work Shift

Data Fields

Name	Data Type	Description
Name	Text	Name of the library.
Description	Text	Textual description of the library. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a library is created.
LibraryList	Sequence of other objects	This is the list of related objects.

Methods

None Defined for Library.

Inherits From: Named Object
Used In: DBAccess

Distribution Statement A: Approved for public release. Distribution is unlimited.

MATERIAL

msmMaterial

This object describes the materials that are used in producing parts. When a new material object is created, its Characteristics and UnitCost objects will automatically be created (but will be empty), and the new material will be put into the appropriate library.

Data Fields

Name	Data Type	Description
Name	Text	Name of the material.
Description	Text	Textual description of the material. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a material is created.
UnitCost	Value With Units	Unit cost in dollars per unit (ex. Dollars per pound).
Type	Text	Material type.
Form	Text	Material form, e.g. bar, sheet
Characteristics	Sequence of Characteristics	Name / Value pairs for additional material characteristics.

Methods

None Defined for Material.

Inherits From: Named Object
Used In: Part

MANUFACTURING ORDER

msmMfgOrder

This object represents a manufacturing order for a quantity of parts. Its primary information is a quantity that can not be directly related to an alternative or process plan. It initiates the execution of a process plan on the shop floor.

Data Fields

Name	Data Type	Description
Name	Text	Name of the manufacturing order.
Description	Text	Textual description of the manufacturing order. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a manufacturing order is created.
Schedule	SchedInfo	Schedule information associated with the order.
Quantity	Integer	Number of parts in an order.

Methods

None Defined for Mfg Order

Inherits From: Named Object
Used In: Process Plan

MANUFACTURING PROGRAM

msmMfgProgram

Thisobject describes a manufacturing program and contains information on production quantity and build rate. It is associated with a specific design study. All Mfg Programs will automatically be added to a database library upon creation.

Data Fields

Name	Data Type	Description
Name	Text	Name of the manufacturing program.
Description	Text	Textual description of the manufacturing program. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a manufacturing program is created.
ProdQty	Integer	Production quantity for the program.
BuildRatePerMonth	Integer	Monthly build rate.
FirstArticleTimeHr	Real No	Time it takes to build first article in hours.
LearningCurveSlope	Real No	Slope of the production learning curve.
TechYear	Integer	Defines the technology readiness year for the program.
AvgArticleTimeHr	Real No	Average time it takes to build a unit.

Methods

None Defined for Manufacturing Program.

Inherits From: Named Object
Used In: Design Alternative

NAMED OBJECT

msmNamedObject

This is a base object for most msm objects. It defines data fields for Name, Description, and Date-Time.

Data Fields

Name	Data Type	Description
Name	**Text**	**Name of the object.**
Description	**Text**	**Textual description of the object. Content is at user's discretion.**
DateTime	Date Time	**Date and time assigned by the system when an object is created.**

Methods

None Defined for Named Object

Inherits From: BaseObject
Used In: **Most msm objects.**

OBJECT SEQUENCE

msmObjectSeq

This object defines a list of similar objects and provides the operations to add and access these objects. Constructor for this object will only be called by other objects within the server which contain msmObjectSeq's. Constructor must control whether duplicate names are allowed, and operations must check as appropriate.

Data Fields

Name	Data Type	Description
DuplicateAllowed	True / False	Indicates whether this sequence allows duplicates.
SeqType	ObjectType	Defines the type of object in this sequence.
NoInSeq	Integer	Number of objects currently in this sequence.
objectSequence	baseObjectSeq	Returns a list of all objects in the sequence
ObjectStructSeq	baseObjectStructSeq	Returns a list of all objects along with their names and descriptions

Methods

Name	Returned Data Type	Parameters (I/O/IO) / Name / Type	Method Description
AddObject	None	(I) / New Obj / Any Obj	Adds object to list.
InsertBefore	None	(I) / New Obj / Any Obj (I) / Ref Obj / Any Obj	Inserts a new object before the reference object.
RemoveObject	None	(I) / Del Obj / Any Obj	Removes this object from list.
FindObject	Any Object	(I) / Name / text (I) / StrtIndex / Integer (O) / Obj Index / Integer	Returns an object from the list, found by name, starting at the given list index.
GetByIndex	Any Object	(I) / Index / Integer	Returns the object located at the given position in the list.

Inherits From: Base Object

Used In: Many MSM objects

OPERATION

msmOperation

This object models an operation, or job step within a process plan. For generality a job step may be another complete process plan. The Cost, Schedule, and Risk objects are automatically created when the Operation is created. The Precedents, Characteristics, PersonResApplic, ToolResApplic, and Features sequences are also automatically created. The list of Characteristics for this operation will be copied from the reference process if it is added. The Characteristics list may be added to manually if desired.

Data Fields

Name	Data Type	Description
Name	Text	Name of the operation.
Description	Text	Textual description of the operation. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a operation is created.
Type	Text	Name of operation type.
Quantity	Integer	Number of repetitions of this operation within Process Plan.
CriticalPath	True/False	Denotes that this operation is on the critical path in this Process Plan.
Runtime	Real Number	Contains a detailed breakdown of the time elements of this operation. Summary time information is to be stored in the Schedule attribute. Actual duration of operation in hours, excluding setup and queue times.
SetupDurationHr	Real Number	Contains a detailed breakdown of the time elements of this operation. Summary time information is to be stored in the Schedule attribute. Duration of the setup in hours.
QueDurationHr	Real Number	Contains a detailed breakdown of the time elements of this operation. Summary time information is to be stored in the Schedule attribute. Time spent in queue in hours.
ProcPlan	Process Plan	This identifies this operation as a nested process plan. If this field is null, this is a discrete operation.
QueueTotalCapacity	Integer	Queue capacity - number of items necessary to fill the queue.
QueueAvgCapacity	Integer	Average number in queue.

Cost	Cost Info	Cost information for this operation.
Schedule	Sched info	Schedule information for this operation.
Risk	Risk Info	Risk information for this operation.
RefProcess	Ref Process	Adding the Reference Process causes its list of OpChar characteristics to be added to the Characteristics sequence below. The reference process associated with an operation that contains standard process information.
Id	Text	ID number / name for this operation.
SetupDescription	Text	Description of the setup activity for this process.
Precedents	Sequence of Operations	List of precedent operations which must be complete prior to starting this operation.
Characteristics	Sequence of Characteristic	User defined name / value pairs which extend the data fields for this operation.
PersonResApplic	Sequence of Resource Applications	List of Personnel type Resource Applications used in this operation.
ToolResApplic	Sequence of Resource Applications	List of Tool type Resource Applications used in this operation.
ConsumedParts	Sequence of PartUsage	List of parts used in this operation.
ProducedParts	Sequence of PartUsage	List of parts produced by this operation.
Features	Sequence of Features	List of part features associated with this operation.

Methods

None Defined for Operation.

Inherits From: Named Object
Used In: Process Plan

PART

msmPart

This object defines a part to be manufactured. It may be a detailed part or an assembly. When a part is created it will automatically create its Cost, Sequence of AssociatedParts, and Features objects.

Data Fields

Name	Data Type	Description
Name	Text	Name of the part.
Description	Text	Textual description of the part. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a part is created.
RejectionRate	Real Number	Rejection rate in percentage of parts that are unacceptable (ex. 0.33)
QuantityPerProduct	Integer	Number of units of a part in the final product.
Type	PartType	Identifies the part type : Detailed or Assembly
Complexity	TextQual	Part complexity: Low, Med, or High
Cost	Cost Info	Cost information for a part.
CADModels	Sequence of CAD Model	Associated CAD models for the part.
Material	Material	Part material.
Number	Text	Part number identifier.
Family	Text	Part family.
AssociatedParts	Sequence of Parts	List of parts in this part, if it is an assembly.
Features	Sequence of Features	List of Features for a detailed part.

Methods

None Defined for Part.

Inherits From: Named Object
Used In: Process Plan , Operation

PART LOCATION

msmPartLocation

This interface represents the location of a part as it is used in a given operation. It is created automatically when the quantity in the msmPartUsage is set. This object may not be created directly by SAVE Data Model clients, but its attributes may be populated by them.

Data Fields

Name	Data Type	Description
Name	Text	Name of the design study. A unique short name of 8 characters is automatically generated from the specified name for use by codes with character limits.
Description	Text	Textual description of the design study. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a design study is created.
TransformMatrix	**4x4 array of real numbers**	Transform matrix used to map resource CAD model into application space.

Methods

None Defined for Part Location

Inherits From: Named Object

Used In: PartUsage

PART USAGE

msmPartUsage

This interface defines the usage of a part. Part Usage Name should be the same as the part number for the part defined in the usage. When a msmPart is added to a msmPartUsage object, the msmPartUsage name attribute is automatically overwritten with the part number from the msmPart object.

Data Fields

Name	Data Type	Description
Name	Text	Name of the design study. A unique short name of 8 characters is automatically generated from the specified name for use by codes with character limits.
Description	Text	Textual description of the design study. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a design study is created.
Quantity	Integer	Number of units of this part used in a given operation.
Part	Part	The part for this usage.
PartLocations	Sequence of PartLocation	The locations (transformation matrices) for this part as used in an operation. The number of locations will be forced to match the quantity attribute if the createloc boolean parameter in the SetQuantity method is set to True. This sequence should only be created/modified through the SetQuantity method to assure consistency.

Methods

Name	Returned Data Type	Parameters (I/O/IO) / Name / Type	Method Description
SetQuantity	None	(I) / newquantity / real number (I) / Boolean / createloc	Sets the quantity attribute. If the createloc parameter is set to True, this method automatically creates the list of part locations, with the number in the list set equal to the quantity attribute.

Inherits From: Named Object

Used In: Operation

PERSONNEL

msmPersonnel

This object defines a personnel resource. It inherits from Resource. Personnel objects are grouped in a Library, and new Personnel are automatically added to this Library.

Data Fields

Name	Data Type	Description
Name	Text	Name of the Personnel resource.
Description	Text	Textual description of the Personnel resource. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a Personnel is created.
Efficiency	Real Number	Decimal percentage of available usage. (ex. 0.23)
CADMod	Sequence of CAD Model	CAD models of the resource.
LaborRate	Real Number	Defines personnel cost - US $ per Hour.
LaborRateYear	Integer	Year for which labor rate is effective.
Calendar	Sequence of Work Calendar	Work calendars for this personnel type.
Shift	Work Shift	Work shift for this personnel type.
Skill	Text	Defines the personnel skill category.

Methods

None Defined for Personnel.

Inherits From: Resource
Used In: Operation

PROCESS PLAN

msmProcessPlan

This is a central object within this model. It contains an ordered set of operations (job steps) which define a manufacturing process. For purposes of generality, a job step in one process plan can be another process plan. A process plan has an associated part (detailed or assembly) and rolls up the cost, schedule, and risk values of its job steps. When a process plan is created, it will automatically create Cost, Schedule, Risk, MfgOrder objects. Characteristics, SimMod, ToolPool, PersonnelPool, and Operations sequences will be created but not populated.

Data Fields

Name	Data Type	Description
Name	Text	Name of the process plan.
Description	Text	Textual description of the process plan. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a process plan is created.
PPStatus	Status	Configuration management flag indicating either Working, In Review, or Released. Status of In Review and Release write lock the process plan. The process plan status flag controls the access status of all objects associated with this plan except reference data typically stored in a library (Break, Breakdown, CAD Model, Inflation Table, Material, Mfg Program, Part, Personnel, Ref Process, Tool, Work Calendar, and Work Shift).
AvgHrCriticalPath	Real Number	Average time for critical path steps, Hours
WaitHr	Real Number	Time spent waiting, Hours. This data field is simply a method to sum all queue durations.
EBOM	Sequence of Parts	Engineering Bill of Materials. Typically the starting point for a planning activity and will not be modified by simulation tools once created.
MBOM	Sequence of Parts	Manufacturing Bill of Materials. This is created as part of the planning process. Many parts will be the same in the MBOM and the EBOM.
Cost	Cost Info	Cost information for this process plan.
Schedule	Schedule Info	Schedule information for this process plan.
Risk	Sequence of Risk Info	Risk information for this process plan.
MfgOrder	Sequence of Mfg Order	Identifies the list of manufacturing order associated with this process plan.

Characteristics	Sequence of Characteristic	List of name / value pairs that can be used to extend the data fields for this process plan.
SimMod	Sequence of Sim Models	List of simulation models used to simulate this process plan. Each simulation code that is executed can record its use and the data files that were used.
ToolPool	Sequence of Resource Pool	List of Tool-type Resource Pools that are available to this process plan.
PersonnelPool	Sequence of Resource Pool	List of Personnel-type Resource Pools that are available to this process plan.
Operations	Sequence of Operations	Ordered list of Operations or job steps in this process plan. Note that an operation may be another process plan, allowing nested, complex plans to be defined.

Methods

None Defined for Process Plan.

Inherits From: Named Object
Used In: Design Alternative

REFERENCE PROCESS

msmRefProcess

This object defines the set of characteristics about a standard manufacturing process. A few general attributes are explicitly defined, and a Characteristics object is included to allow other attributes to be defined for the reference process. A second Characteristics object is included to define the characteristics needed by the Operation object which is of this Reference Process type. Reference Processes are automatically placed in a Library when they are created. When a Reference Process is created it will also create its associated Sequences of Risk, Characteristics, and Operations Characteristics.

Data Fields

Name	Data Type	Description
Name	Text	Name of the reference process.
Description	Text	Textual description of the reference process. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a reference process is created.
Stdhrs	Real Number	Value from historical data for the standard hours it takes to complete this process.
Stability	TextQual	Qualitative process stability : Low, Med, High
Maturity	TextQual	Qualitative process maturity: Low, med, High
Complexity	NumQual	Qualitative process complexity : 0 - 10
Risk	Sequence of Risk Info	Risk values associated with this process.
Characteristics	Sequence of Characteristics	List of name / value pairs that can be used to extend the data fields of this reference process. This list applies to the generic reference process.
OpChar	Sequence of Characteristics	This list of name / value pairs is added to a specific Operation's Characteristics when it is associated with this reference process.

Methods

None Defined for Reference Process.

Inherits From: Named Object
Used In: Operation

RESOURCE

msmResource

This interface represents a resource utilized by a manufacturing process operation. This is object defines data fields and methods that are common to all types of resources (personnel and tools) and will have Personnel and Tool objects derived from it. Constructor will create CadMod and initialize Efficiency to 0.

Data Fields

Name	Data Type	Description
Name	Text	Name of the resource.
Description	Text	Textual description of the resource. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a Resource is created.
Efficiency	Real Number	Decimal percentage of available usage. (ex. 0.23)
CADMod	Sequence of CAD Model	List of CAD models of the resource.

Methods

None Defined for Resource

Inherits From: Named Object
Inherited by: Personnel, Tool

RESOURCE APPLICATION

msmResourceApplic

This object represents the utilization of a resource in one operation. It is used to specify the location of the resource in the factory model and the percentage of the available resource that is used in this one operation. Utilization is set to 0 when it is created.

Data Fields

Name	Data Type	Description
Name	Text	Name of the resource application.
Description	Text	Textual description of the resource application. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a resource application is created.
Quantity	Integer	Quantity of the resource used in this application.
TransformMatrix	Matrix	Transformation matrix use to map resources CAD model into application space (typedef float Matrix [4][4];).
Pool	Resource Pool	Pool from which this resource is drawn. The available pools are listed in the associated Process Plan.

Methods

Name	Returned Data Type	Parameters (I/O/IO) / Name / Type	Method Description
SetQuantity	None	(I) / New Quantity / Real	Sets original or new quantity of this resource.

Inherits From: Named Object
Used In: Operation

RESOURCE POOL

MsmResourcePool

This object defines a pool of one type of resource which will track both the total quantity of this resource that is available to a Process Plan and also will keep track of the total utilization of the resources in this pool by the Process Plan. The ResouceUitilization data structure will reference this pool to track the utilization of a resource at the Operation level. Utilization is set to 0.0 when this object is created.

Data Fields

Name	Data Type	Description
Name	Text	Name of the resource pool.
Description	Text	Textual description of the resource pool. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a resource pool is created.
Quantity	Real Number	Quantity of the resource in this pool. Fractional quantities are allowed.
Utilization	Real Number	Number of individual resources currently in use from this pool. Cannot exceed Quantity.
Resource	Personnel or Tool Resource	Resource object for this pool

Methods

Name	Returned Data Type	Parameters (I/O/IO) / Name / Type	Method Description
SetQuantity	None	(I) / New Quantity / Real No	Sets original or new quantity for this pool.
ChangeUtil	None	(I) / New Utilization / Real No	Changes the utilization - should be called only by ResourceApplication object.

Inherits From: Named Object
Used In: Process Plan, Resource Application

RISK INFORMATION

msmRiskInfo

This object contains the basic set of risk data used throughout the model. All values are stored as versioned float (real) variables.

Data Fields

Name	Data Type	Description
Name	Text	Name of the reference process.
Description	Text	Textual description of the reference process. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a reference process is created.
RiskYield	Real Number	Decimal percentage of correct results produced by a process. (ex. 0.23)
FailureProbability	Versioned Float	Likelihood of the failure of a given activity as percentage defects.
FailureConsequence	Versioned Float	Consequence of failure - provides indication of the level of importance of a failure.
Mean	Versioned Float	Mean value of the risk distribution.
StdDeviation	Versioned Float	Standard deviation - measure of the dispersion of a frequency distribution with respect to the mean.
Cp	Versioned Float	Process capability measurement.
Cpk	Versioned Float	Process capability measurement.
Description	Text	Description of risk item.
QualitativeResults	Text	Textual representation of the basis and assumptions involved in a particular risk analysis.
Contributors	Sequence of Contributor	List of Contributors to this risk and their percentage contribution.

Methods

None Defined for Risk information.

Inherits From: Named Object

Used In: Process Plan, Operation, Reference Process

SCHEDULE INFORMATION

msmSchedInfo

This object contains the basic set of schedule data used throughout the model. All date values are stored as versioned strings **in database standard date/time format (yyyy/mm/dd 24:mm:ss).**

Data Fields

Name	Data Type	Description
Priority	TextQual	Qualitative priority - Low, Med, High
PalannedDurationHr	Versioned Float	Planned duration in hours for a particular task.
ActualDurationHr	Versioned Float	Actual duration in hours for a particular task.
PlannedStartDate	Versioned String	Planned start date for this task or event.
PlannedEndDate	Versioned String	Planned end date for this task or event.
ActualStartDate	Versioned String	Actual start date for this task or event.
ActualEndDate	Versioned String	Actual end date for this task or event.

Methods

None Defined for Schedule Information

Inherits From: Base Object
Used In: Process Plan, Operation

SIMULATION MODEL

msmSimModel

This interface provides a mechanism to store information about the simulations performed for a specific alternative process plan. It includes the simulation software type and name as well as information about the results or output files. (This object stores results - use msmSimReqst to set up simulation code launch information).

Data Fields

Name	Data Type	Description
Name	Text	Name of the sim model.
Description	Text	Textual description of the simulation model. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a sim model is created.
Type	SimType	Category of simulation (e.g. Cost, Factory, …)
Factory	Sequence of CAD Model	Geometry models of the associated factory.
SimCode	Text	Identifies the simulation code, with version, that was used for a simulation execution.
DataLocation	Text	Location of data used for this execution - multiple datasets can be defined in comma delimited list.

Methods

None Defined for Simulation Model.

Inherits From: Named Object
Used In: Process Plan

SIMULATION REQUEST

msmSimReqst

This object contains the primary entry points into the data model that might be needed by a simulation code. It is expected that this will be passed to the simulators by the work flow manager. The attributes InputFiles, InputOptions and OutputOptions can be used to pass a simulator information on launch, input, and output options. These attributes use tab-delimited strings to separate the options.

Typically the trade study team will create these SimReqst objects and include a reference to them in the work flow being developed. The workflow manager passes this reference to the simulator before launch. The simulator wrapper then accesses the SimReqst object to obtain launch options and database context information. Each simulator will determine the syntax of information in these strings.

Data Fields

Name	Data Type	Description
Name	Text	Name of the simulation request.
Description	Text	Textual description of the simulation request. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a sim request is created.
DesignStudy	DesignStudy	DesignStudy for which simulation is to be run.
DesignAlternative	Design Alternative	Design Alternative for which simulation is to be run.
ProcessPlan	Process Plan	Process Plan for which simulation is to be run.
InputFiles	Text	Defines optional input files and/or launch instructions for the simulation code execution.
InputOptions	Text	Strings defining which simulation code input options are to be exercised in this code execution.
OutputOptions	Text	String defining which simulation code output options are to be exercised in this code execution.

Methods

None Defined for Simulation Request.

Inherits From: Named Object

Used In: DBAccess

TOOL

msmTool

Defines a tool type of resource. This interface inherits from Resource. When a new Tool is created, it will automatically create its Characteristics and Breakdown objects and it will automatically be placed into the Tool Library.

Data Fields

Name	Data Type	Description
Name	Text	Name of the Tool.
Description	Text	Textual description of the Tool. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a Tool is created.
Efficiency	Real Number	Decimal percentage of available usage. (ex. 0.23)
CADMod	Sequence of CAD Model	CAD models of the resource.
Cost	Real Number	Cost of the tool in US dollars.
ToleranceCap	Real Number	Tolerance capability for the tool - Inches
FailureRate	Real Number	Tool failure rate - decimal percentage of downtime. - e.g. 0.23
Type	Text	Type of tool (e.g. conveyer, controlled, hand)
Characteristics	Sequence of Characteristics	List of name / value pairs that can be used to extend the data fields in the tool object.
Breakdowns	Sequence of Breakdowns	List of possible tool Breakdowns.

Methods

None Defined for Tool.

Inherits From: Resource
Used In: Resource Pool

VALUE WITH UNITS

msmValueWithUnits

Initially, ValueWithUnits will not be very functional. Calling CheckUnits will always return a Boolean True value and calling Convert Value will return as a float whatever is loaded into the Value attribute. We will implement functionality soon. This is a shortcut to getting a Characteristic **object implemented.**

Data Fields

Name	Data Type	Description
Value	Real Number	This is the attribute value.
Units	Text String	This string defines the Units for the value.

Methods

Name	Returned Value Type	Parameters (I/O/IO) / Name / Type	Method Description
ConvertValue	Real Number	(I) / New Units / Text	Validates the New Units text string and then converts and returns the converted value.
CheckUnits	True / False	(I) / Unit / Text	Validates a unit string as being a valid units string that is compatible with the stored value units.

Inherits From: Base Object
Used In: Characteristic

VERSIONED FLOAT

msmVersionedFloat

This object implements a fine-grained versioning for variables of type float. All versions of a value are retained along with Date and Source of values. A read-only attribute is provided for easy access to the current value. Previous values can be retrieved by date.

Data Fields

Name	Data Type	Description
CurrValue	Real Number	Current numeric value.
CurrSource	Text	Simulation code / version which generated the current value.
CurrDateTime	Text String	Date / time when current value was generated. Standard date-time format: yyyy/mm/dd 24:mm:ss
VersionedFloatStructSeq	VersFloatStructSeq	Provides a sequence of the versioned floats, including the values, sources, and datetimes.

Methods

Name	Returned Data Type	Parameters (I/O/IO) / Name / Type	Method Description
AddNewValue	None	(I) / Val / Real Number (I) / Source / Text	Adds a new value with its source, and associates a current date-time.
GetDatedValue	Real Number	(i) / DateTime / Text	Returns a value that was current at the requested time. DateTime is in standard format: yyyy/mm/dd 24:mm:ss
GetDatedInfo	Real Number	(I) / DateTime / Text (O) / Source / Text	Returns a value that was current at the requested time. Also returns a parameter that defines the value's source. DateTime is in standard format: yyyy/mm/dd 24:mm:ss

Inherits From: Base Object

Used In: Cost, Risk, Schedule

VERSIONED STRING

msmVersionedString

This object implements a fine-grained versioning for variables of type string (text). All versions of a value are retained along with Date and Source of values. A read-only attribute is provided for easy access to the current value. Previous values can be retrieved by date.

Data Fields

Name	Data Type	Description
CurrValue	Text	Current text (string) value.
CurrSource	Text	Simulation code / version which generated the current value.
CurrDateTime	Text String	Date / time when current value was generated. Standard date-time format: yyyy/mm/dd 24:mm:ss
VersionedStringStructSeq	VersStringStructSeq	Provides a sequence of all versions including value, source, and datetime.

Methods

Name	Returned Data Type	Parameters (I/O/IO) / Name / Type	Method Description
AddNewValue	None	(I) / Val / Text (I) / Source / Text	Adds a new value with its source, and associates a current date-time.
GetDatedValue	Text	(i) / DateTime / Text	Returns a value that was current at the requested time. DateTime is in standard format: yyyy/mm/dd 24:mm:ss
GetDatedInfo	Text	(I) / DateTime / Text (O) / Source / Text	Returns a value that was current at the requested time. Also returns a parameter that defines the value's source. DateTime is in standard format: yyyy/mm/dd 24:mm:ss

Inherits From: Base Object
Used In: Schedule

WORK CALENDAR

msmWorkCalendar

This object defines the calendar of available work days for a resource. Work Calendars are automatically stored for use in a Library when they are created.

Data Fields

Name	Data Type	Description
Name	Text	Name of the work calendar.
Description	Text	Textual description of the work calendar. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a work calendar is created.
WorkYear	Integer	Year for the Work Calendar.
Jan1	Days Of Week	Day of the week on which January 1 falls.
NoWorkDays	Integer	Number of days that are not weekend or holidays.

Methods

Name	Returned Value Type	Parameters (I/O/IO) / Name / Type	Method Description
AddDay	None	(i) / Date / Text(DateTime)	Makes a day a work day.
DelDay	None	(i) / Date / Text(DateTime)	Makes a day a non-work day.
WorkDaysBetweenDates	Integer	(i) / Start / Text(DateTime) (I) / Stop / Text(DateTime)	Increment between 2 years must be done in 2 different calls. Returns number of workdays between dates.
IsWorkDay	True/False	(I) / Date / Text(DateTime)	Determines if a given day is a scheduled work day.

Inherits From: Named Object
Used In: Personnel

WORK SHIFT

msmWorkShift

This object defines a working shift to be used in work calendars.

Data Fields

Name	Data Type	Description
Name	Text	Name of the work shift.
Description	Text	Textual description of the work shift. Content is at user's discretion.
DateTime	Date Time	Date and time assigned by the system when a work shift is created.
StartTime	Real Number	Start time - hours after midnight (0-24).
EndTime	Real Number	Shift end time - hours after midnight (0-24).
BreakList	Sequence of Breaks	List of breaks associated with this shift.
HrInShift	Real Number	Number of productive hours in shift.

Methods

None Defined for Work Shift.

Inherits From: Named Object
Used In: Personnel

Appendix D

ASURE Wrapper User's Guide

SAIC

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

1.0 ASURE Interface for SAVE

The ASURE SAVE interface consists of:

(1) Startup executable “AsureStart.exe”;

(2) Simulation server named “AsureServer.exe”

(3) Four executable clients in the Externals folder named “importDB.exe”, “exportDB.exe”, “exAttrDB.exe”, and “imAttrDB.exe”.

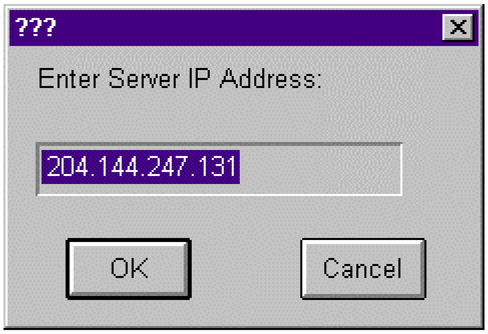
(4) ASURE data files “\data\alternatives.dat” and “\Externals\Server.dat”.

1.1 Installation

ASURE software is based on Wingz® and requires its installation prior to use.

1.2 Setup/Configuration

Launch the ASURE software by executing “AsureStart.exe”. Configure the SAVE Database Server IP Address by selecting <Import/Export> menu item and <Configure>. Enter the Server IP Address and select <OK>. The Server IP Address configuration is then saved to the “Server.dat” file in the Externals folder.



Enter Server IP Address:

204.144.247.131

2.0 SAVE Database Interactions

ASURE import and export operations to the SAVE database are accomplished by first populating the Simulation Request “SimReqst” object with the SAVE Query Manager “QM”.

2.1 Importing Process Plans

Currently ASURE’s initial transaction with the SAVE database requires the user to perform a two step process.

1. First select the <Import/Export> menu and the <Import> menu item and choose <Request Data>. The user then is prompted to “Enter Simulation Request Name”. Enter in the simulation request and select <OK>. Next the user is prompted to “Enter Process Plan Name” with the default name shown. Enter the Process Plan name and select <OK>. After

entering the simulation request and Process Plan names, the ASURE data interface retrieves the specific Process Plan data from the SAVE database and writes the data to an output file. The file name is automatically generated from the Process Plan name (ProcessPlanName.dat) and is stored locally in the ASURE data folder.

2. Now to import the Process Plan data into ASURE, the user then selects the <Import/Export> menu and the <Import> menu item and choose <Design Data>. The user is then prompted to enter the Process Plan data file name with extension (ProcessPlanName.dat). Then ASURE imports the Process Plan data hierarchy and displays it to the screen, with a prompt to update the ASURE database. Once the user has selected to update the ASURE database, the data import is transacted into the local ASURE file system.

2.2 Importing Attribute Data

Navigate to a Process Plan or Operation description sheet. Select an import variable and the <Import/Export> menu and the <Import> menu item and choose <Attribute Data>. The ASURE data interface then retrieves the current data from the SAVE database or returns an error if the object is not found.

2.3 Exporting Attribute Data

Navigate to a Process Plan or Operation description sheet. Select an export variable and the <Import/Export> menu and the <Export> menu item and choose <Attribute Data>. The ASURE data interface then retrieves the current data from the SAVE database or returns an error if the object is not found.

2.4 Exporting Process Plan Data

The ASURE export Process Plan data transaction with the SAVE database requires the user to perform a two step process.

1. First select the <Import/Export> menu and the <Export> menu item and choose <Design Data>. The user is then prompted to enter the Process Plan data file name with extension (ProcessPlanName.dat). Then the ASURE data interface exports the Process Plan data to the ASURE file system and messages upon completion.

2. Second to export the process plan data to the SAVE database, the user then selects the <Import/Export> menu and the <Export> menu item and choose <Submit Data>. The user then is prompted to "Enter Simulation Request Name". Enter in the simulation request and select <OK>. Next the user is prompted to "Enter Process Plan Name" with the default name shown. Enter the Process Plan name and select <OK>. After entering the Simulation Request and Process Plan names, the ASURE data interface exports the specific Process Plan data from the ASURE file system to the SAVE database.

3.0 ASURE ProcessTest Alternative

An example alternative “ProcessTest” has been created with the Simulation Request “Learning Aide” and the Process Plan name “ProcessTest”. The Process Plan description for “ProcessTest” and the Operation “inspct01” have example description sheets saved in this release of ASURE.

The names for Process Plan and Operation description attributes follow. These names are automatically generated, if data exists, when a Process Plan is imported is into ASURE. These names are for local ASURE use only and allow the user to group all Process or Operation relevant information on one description sheet.

Table D-1: Process Plan Description Attributes

Name	**SAVE Database Object**
<OperationName>	Operation Name
Description	Process Plan Description
Status	Process Plan Status
PlanDur	Schedule Planned Duration
PlanEnd	Schedule Planned End Date
PlanStart	Schedule Planned Start Date
<manufactureOrderName>_mdesc	Manufacture Order Name
<manufactureOrderName>_moqty	Manufacture Order Qty
<characteristicName>_char	Characteristic Name
<partName>_rrate	Part MBOM Rejection Rate
<partName>_<featureName>_fqty	Part MBOM Feature Qty
<partName>_<featName>_<charName>_fchar	Part MBOM Feature Characteristic
<riskName>_cp	Risk Cp
<riskName>_cpk	Risk Cpk
<riskName>_risk	Risk hi/lo from (mean ± std)
<riskName>_rdesc	Risk Description
<riskName>_fprob	Risk Probability of Failure
<riskName>_yld	Risk Yield
<riskcontributorName>_rcon	Risk Contributor Percent

Table D-2: Operation Description Attributes

Name	**SAVE Database Object**
<processName>	Nested ProcessPlan Name
Description	Operation Description
PlanDur	Schedule Planned Duration
PlanEnd	Schedule Planned End Date
PlanStart	Schedule Planned Start Date
<characteristicName>_char	Characteristic Name
<featureName>_fqty	Feature Qty
<featureName>_<charName>_fchar	Feature/Characteristic
<partuseName>_cqty	PartUsage(Consumed) Qty
<partuseName>_pqty	PartUsage(Produced) Qty
<partuseName>_<partName>_crate	PartUsage(Consumed) Rejection Rate
<partuseName>_<partName>_prate	PartUsage(Produced) Rejection Rate
<partuseName>_<partName>_<featName>_cqty	PartUsage(Consumed)/Part/Feature Qty
<partuseName>_<partName>_<featName>_pqty	PartUsage Produced/Part/Feature Qty
<partuseName>_<partName>_<featName>_<charName>_cfchar	Part Usage(Consumed)/Part/Feature/Char
<partuseName>_<partName>_<featName>_<charName>_pfchar	Part Usage(Produced)/Part/Feature/Char
<partuseName>_<partName>_<matName>_<charName>_cmat	PartUsage(Comsumed)/Part/Material Description
<partuseName>_<partName>_<matName>_<charName>_pmat	PartUsage(Produced)/Part/Material Description
<partuseName>_<partName>_<matName>_<charName>_cfrm	PartUsage(Comsumed)/Part/Material Form

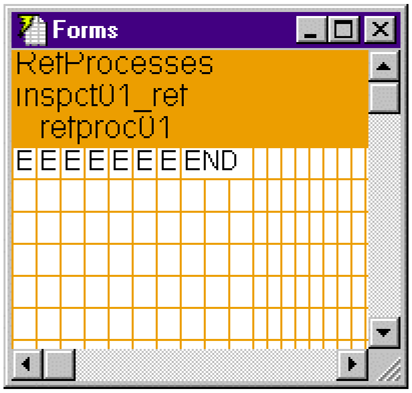
Name	**SAVE Database Object**
<partuseName>_<partName>_<matName>_<charName>_pfrm	PartUsage(Produced)/Part/Material Form
<partuseName>_<partName>_<matName>_ctyp	PartUsage(Comsumed)/Part/Material Type
<partuseName>_<partName>_<matName>_ptyp	PartUsage(Produced)/Part/Material Type
<riskName>_cp	Risk Cp
<riskName>_cpk	Risk Cpk
<riskName>_risk	Risk hi/lo from (mean ± std)
<riskName>_rdesc	Risk Description
<riskName>_fprob	Risk Probability of Failure
<riskName>_yld	Risk Yield
<riskcontributorName>_rcon	Risk Contributor Percent

Table D-3: Reference Process Description Attributes

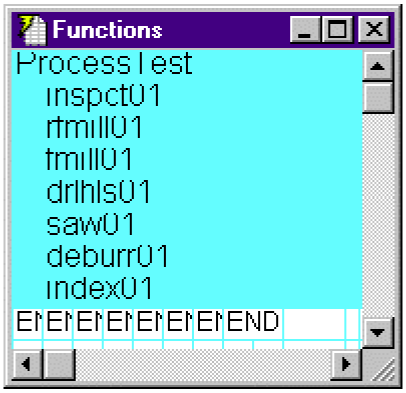
Name	SAVE Database Object
Description	Reference Process Description
maturity	Reference Process Maturity
complexity	Reference Process Complexity
<riskName>_cp	Risk Cp
<riskName>_cpk	Risk Cpk
<riskName>_risk	Risk hi/lo from (mean ± std)
<riskName>_rdesc	Risk Description
<riskName>_fprob	Risk Probability of Failure
<riskName>_yld	Risk Yield
<riskcontributorName>_rcon	Risk Contributor Percent
<characteristicName>_char	Reference Process Characteristic

3.1 "ProcessTest" Alternative Forms and Functions

Reference Processes are imported from the SAVE database into ASURE and are represented in the Forms hierarchy. The Process Plan and operations are shown in the Functions hierarchy. A Reference Process name is denoted as a child of an entry with "_ref" extension. The Operation "inspct01" has a Reference Process named "refproc01". Reference Process data for "refproc01" is stored in its description sheet. Sample description sheets follow for Process Plan, Operation, and Reference Processes.

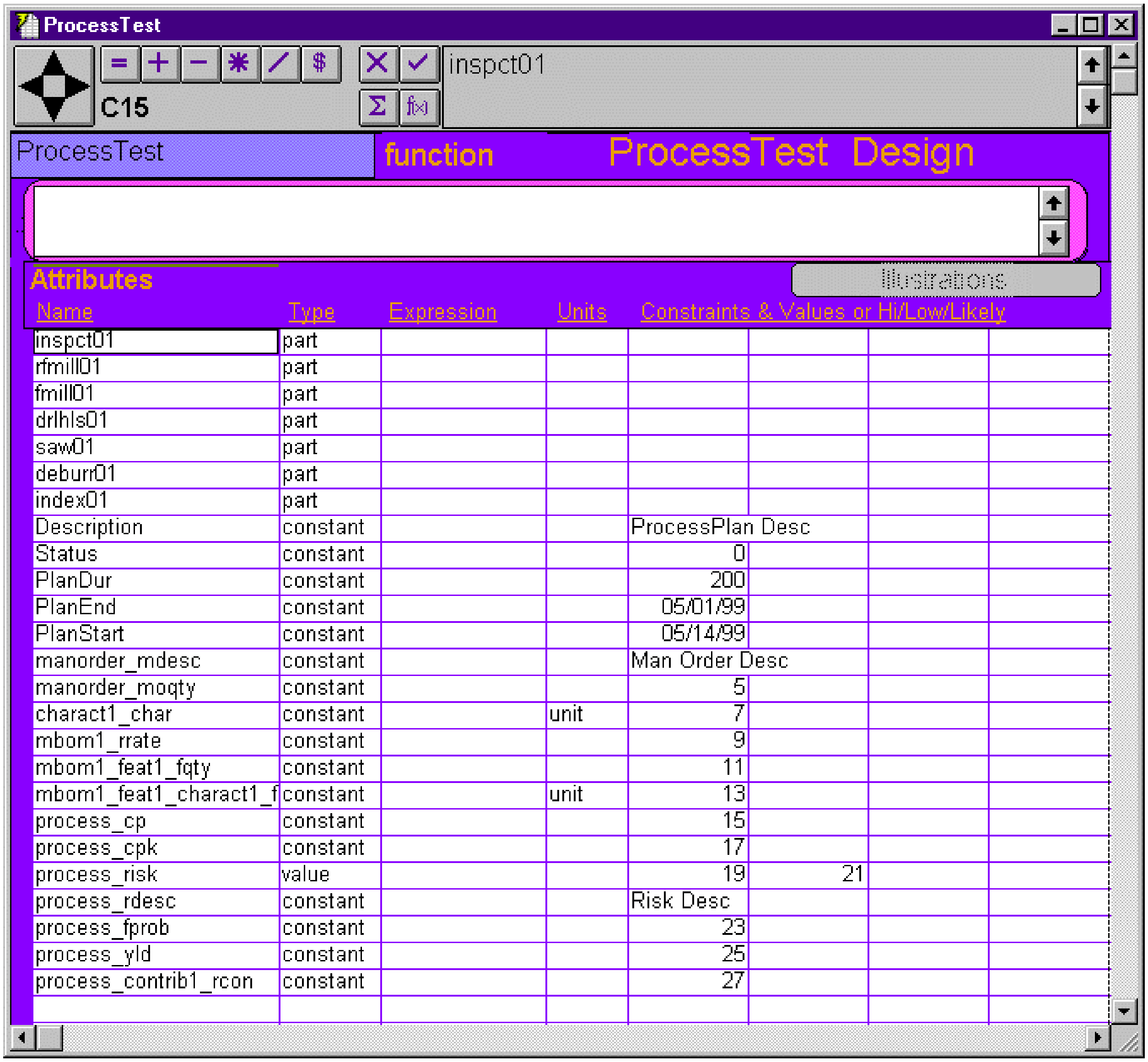


```
RefProcesses
inspct01_ref
  refproc01
E E E E E E E END
```



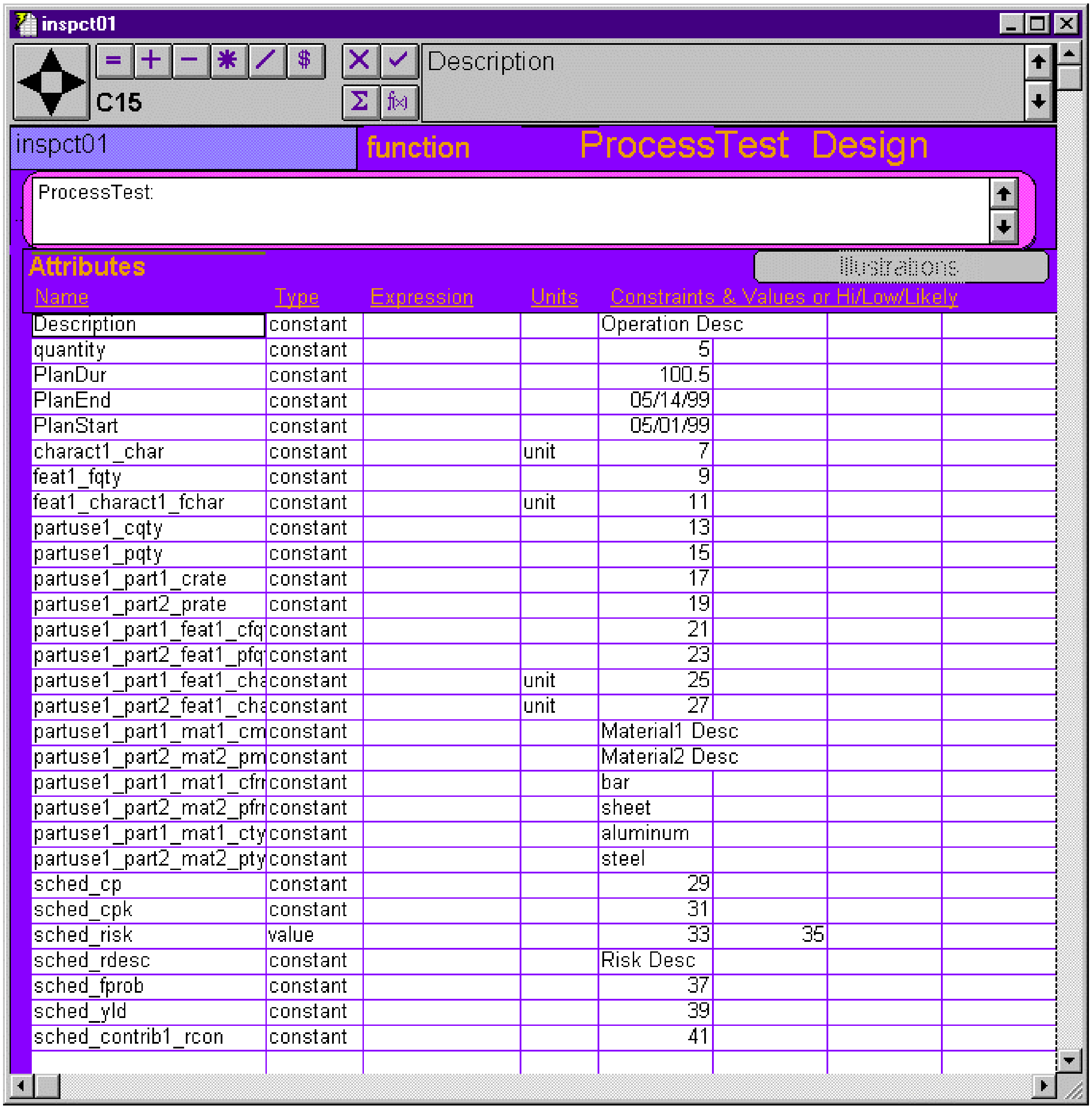
```
ProcessTest
  inspct01
  rtmill01
  tmill01
  drlhls01
  saw01
  deburr01
  index01
EEEEEEEEND
```

Process Plan Description Sheet



Name	Type	Expression	Units	Constraints & Values or Hi/Low/Likely			
inspct01	part						
rfmill01	part						
fmill01	part						
drlhls01	part						
saw01	part						
deburr01	part						
index01	part						
Description	constant			ProcessPlan Desc			
Status	constant			0			
PlanDur	constant			200			
PlanEnd	constant			05/01/99			
PlanStart	constant			05/14/99			
manorder_mdesc	constant			Man Order Desc			
manorder_moqty	constant			5			
charact1_char	constant		unit	7			
mbom1_rrate	constant			9			
mbom1_feat1_fqty	constant			11			
mbom1_feat1_charact1_f	constant		unit	13			
process_cp	constant			15			
process_cpk	constant			17			
process_risk	value			19	21		
process_rdesc	constant			Risk Desc			
process_fprob	constant			23			
process_yld	constant			25			
process_contrib1_rcon	constant			27			

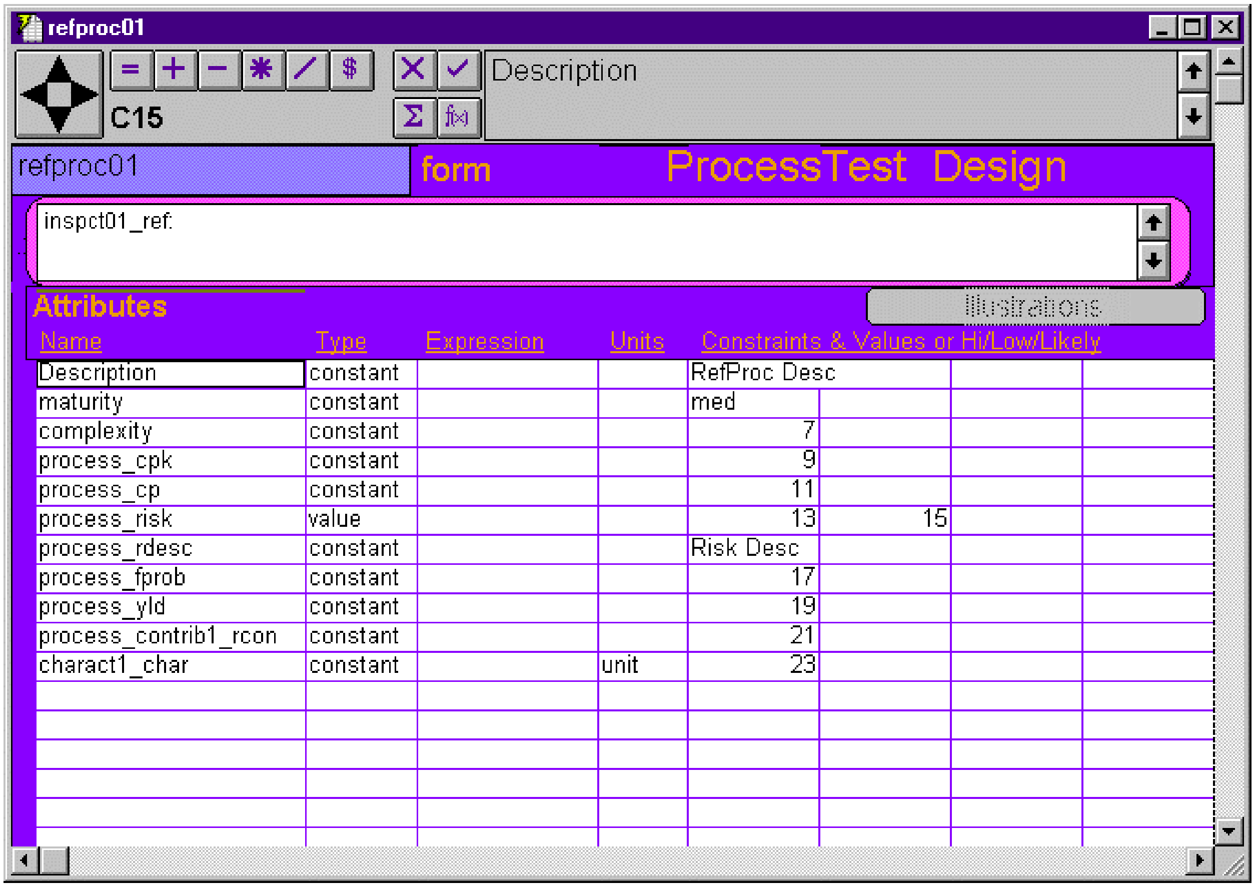
Operation Description Sheet



ProcessTest:

Name	Type	Expression	Units	Constraints & Values or Hi/Low/Likely			
Description	constant			Operation Desc			
quantity	constant			5			
PlanDur	constant			100.5			
PlanEnd	constant			05/14/99			
PlanStart	constant			05/01/99			
charact1_char	constant		unit	7			
feat1_fqty	constant			9			
feat1_charact1_fchar	constant		unit	11			
partuse1_cqty	constant			13			
partuse1_pqty	constant			15			
partuse1_part1_crate	constant			17			
partuse1_part2_prate	constant			19			
partuse1_part1_feat1_cfq	constant			21			
partuse1_part2_feat1_pfq	constant			23			
partuse1_part1_feat1_cha	constant		unit	25			
partuse1_part2_feat1_cha	constant		unit	27			
partuse1_part1_mat1_cm	constant			Material1 Desc			
partuse1_part2_mat2_pm	constant			Material2 Desc			
partuse1_part1_mat1_cfr	constant			bar			
partuse1_part2_mat2_pfr	constant			sheet			
partuse1_part1_mat1_cty	constant			aluminum			
partuse1_part2_mat2_pty	constant			steel			
sched_cp	constant			29			
sched_cpk	constant			31			
sched_risk	value			33	35		
sched_rdesc	constant			Risk Desc			
sched_fprob	constant			37			
sched_yld	constant			39			
sched_contrib1_rcon	constant			41			

Reference Process Description Sheet



C15

Description

refproc01

form

ProcessTest Design

inspct01_ref:

Attributes

Name	Type	Expression	Units	Constraints & Values or Hi/Low/Likely			
Description	constant			RefProc Desc			
maturity	constant			med			
complexity	constant			7			
process_cpk	constant			9			
process_cp	constant			11			
process_risk	value			13	15		
process_rdesc	constant			Risk Desc			
process_fprob	constant			17			
process_yld	constant			19			
process_contrib1_rcon	constant			21			
charact1_char	constant		unit	23			

Distribution Statement A: Approved for public release. Distribution is unlimited.

4.0 Import/Export

4.1 Import

Import interface to the SAVE database.

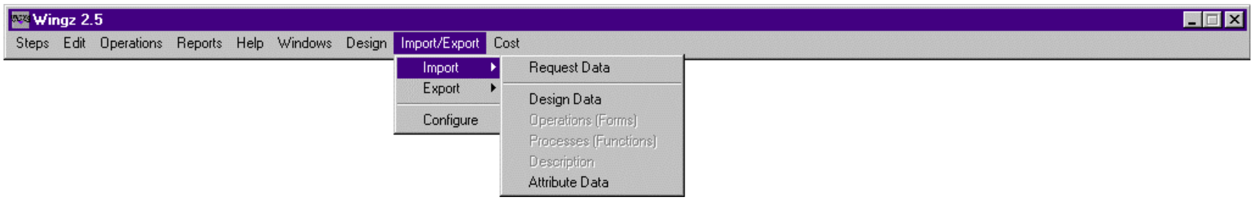


Figure D-1: The ASURE Import Sub Menu

Request Data—Entering the simulation request and the process plan name.

Design Data—Importing previously design data from the ASURE I/O file database.

Attribute Data—Importing attribute data from the SAVE database.

4.2 Export

Export interface to the SAVE database.

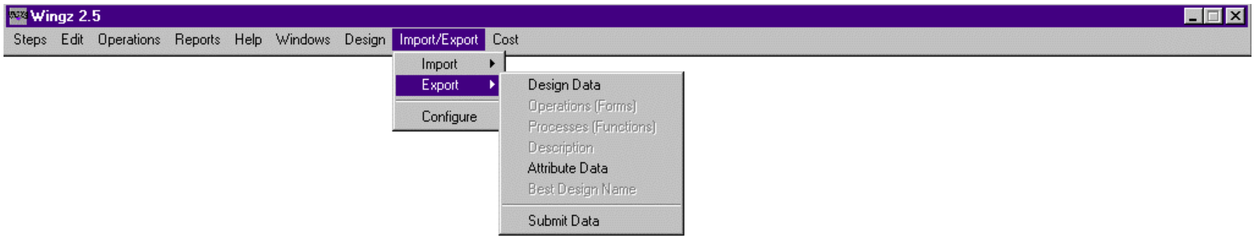


Figure D-2: The ASURE Export Sub Menu

Design Data—Exporting design data to the ASURE I/O file database.

Attribute Data—Exporting attribute data to the SAVE database.

Submit Data—Submitting data to the SAVE database.

4.3 Configure

Configuring the SAVE database Server IP Address.

5.0 Process Plan Object Hierarchy

The capabilities of the ASURE wrapper are indicated in the table below.

Process Plan
Date Time
Description
Name
Status
Process Plan/Characteristics
Name
Numerical Value
Process Plan/Cost/Inflation Table
Description
Name
Quantity
Description
Name
Quantity
Name
Numerical Value
Process Plan/Operation/ Features
Name
Quantity
Process Plan/Operation/Features/ Characteristics
Name
Numerical Value
Process Plan/Operation/Part Usage
Name
Quantity
Process Plan/Operation/Part Usage/Part
Name
Rejection Rate
Process Plan/Operation/Part Usage/Part/Feature
Name
Quantity
Process Plan/Operation/Part Usage/Part/Feature/Characteristics
Name
Numerical Value

ProcessPlan/Operation/Part Usage/Part/Material
Description
Form
Name
Type
Process Plan/Operation/ Reference Process
Complexity
Maturity
Name
Process Plan/Operation/ Reference Process/ Characteristics
Name
Numerical Value
Process Plan/Operation/ Reference Process/Risk
Cp
Cpk
Description
Mean
Name
Probability of Failure
Standard Deviation
Yield
Process Plan/Operation/ Reference Process/Risk/ Contributors
Name
Percent Contribution
Process Plan/Operation/Risk
Cp
Cpk
Description
Mean
Name
Probability of Failure
Standard Deviation
Yield
Process Plan/Operation/Risk/Contributors
Name
Percent Contribution

Process Plan/Operation/ Schedule
Planned Duration
Planned End Date
Planned Start Date
Process Plan/Part (MBOM)
Name
Rejection Rate
Process Plan/Part (MBOM)/Feature
Name
Quantity
Process Plan/Part (MBOM)/ Feature/ Characteristics
Name
Numerical Value
Process Plan/Risk
Cp
Cpk
Description
Mean
Name
Probability of Failure
Standard Deviation
Yield
Process Plan/Risk/ Contributors
Name
Percent Contribution
Process Plan/Schedule
Planned Duration
Planned End Date
Planned Start Date

6.0 References

Wingz, Working with Wingz, Informix Software, Inc., Lenexa, KS, 1993.

Wingz HyperScript Manual, Informix Software, Inc., Menlo Park, CA, 1992.

Wingz Reference Manual, Informix Software, Inc., Menlo Park, CA, 1992.

Wingz User's Guide, Informix Software, Inc., Menlo Park, CA, 1992.

Appendix E

Cost Advantage Wrapper User's Guide

Cognition Corporation

SAVE Software User's Manual
Contract Number F33615-95-C-5538
58
CDRL A012

1.0 The Import Client

The import client extracts data from the given Simulation Request present in the SAVE database and builds a cost note in a Cost Advantage (CA) Session. The client executable is named client_in. The SAVE operations are imported as CA assembly features. If the operation has a process plan, a new part is added to CA instead of an assembly feature. If the part type is Detail, the CA part type becomes component, and the part features are added as CA features.

The following nomenclature is used relative to SAVE and CA.

1.1 SAVE Notation

SimRequest

The name of the Simulation Request. This is the entry point into the SAVE database.

DesignAlternative

The design alternative of the simulation request.

ProcessPlan

The ProcessPlan for a simulation request.

Operation

The operations within a Process Plan. If a process plan is associated with an operation, the operation becomes a nested process plan.

Part

The part associated with a process plan.

Features

The features within a part.

1.2 CA Notation

Cost Note

The top level object in a CA session.

Assembly/Component

If the SAVE part type is detail, the cost note is of type Component. Other-wise, the cost note is of type Assembly.

Part

If the cost note is of type Component, there is only one root part in CA of type Component. If the cost note type is Assembly, there can be nested parts in CA.

Assembly Features / Cha racteristics

The features within a cost note/part of type Assembly.

Features / Feature Characteristics

The features within a cost note/part of type Component.

Process / Process Characteristics

The Process object and the characteristics of a part.

Material / Material Characteristics

The Material object and the characteristics of a part of type Component.

1.3 The Map File

The map file is used for filtering and mapping from the SAVE objects to CA objects. A template map file which does a generic mapping is provided in the "map.save.import.template" file. The template file should be copied as the file "map.save.import" within each cost model directory and edited as appropriate. The entries that have "#" in the first column are treated as comments. Each entry in the map file must be on a single line (should not wrap). The entries in the map file are of three distinct types:

1. Entries that are directives telling whether to import data from particular objects or not. These entries are paths to be traversed in the SAVE data model to get to the particular object. Each path contains a set of tokens or keywords which are the actual names of the objects and attributes of the SAVE model sans the 'msm' prefix. The tokens or keywords are separated by semicolons (;).

The following entries correspond to the process plan, feature, material, and operation names within the SAVE objects:

<process_name>, <feature_name>, <material_name>, <op_name>

The values for these can be exact names, or can be regular expressions.

```
Imp;ProcessPlan;

Imp;ProcessPlan;<process_name>;

Imp;ProcessPlan;<process_name>;Characteristic;

Imp;ProcessPlan;<process_name>;MfgOrder;

Imp;ProcessPlan;<process_name>;Part;

Imp;ProcessPlan;<process_name>;Part;CADModel;
```

Imp;ProcessPlan;<process_name>;Part;Feature;

Imp;ProcessPlan;<process_name>;Part;Feature;<feature_name>;

Imp;ProcessPlan;<process_name>;Part;Feature;<feature_name>;Characteristics;

Imp;ProcessPlan;<process_name>;Part;Material;

Imp;ProcessPlan;<process_name>;Part;Material;<material_name>;

Imp;ProcessPlan;<process_name>;Part;Material;<material_name>;Characteristic;

Imp;ProcessPlan;<process_name>;ResourcePool;Resource;

Imp;ProcessPlan;<process_name>;ResourcePool;Resource;CADModel;

Imp;ProcessPlan;<process_name>;ResourcePool;Resource;Personnel;

Imp;ProcessPlan;<process_name>;ResourcePool;Resource;Tools;

Imp;ProcessPlan;<process_name>;ResourcePool;Resource;Tools;

Imp;ProcessPlan;<process_name>;ResourcePool;Resource;Tools;Characteristic;

Imp;ProcessPlan;<process_name>;Risk;

Imp;ProcessPlan;<process_name>;Schedule;

Imp;DesignAlternative;

Imp;DesignAlternative;MfgProgram;

Imp;ProcessPlan;<process_name>;Operation;

Imp;ProcessPlan;<process_name>;Operation;<op_name>;

Imp;ProcessPlan;<process_name>;Operation;<op_name>;Characteristics;

Imp;ProcessPlan;<process_name>;Operation;<op_name>;Risk;

Imp;ProcessPlan;<process_name>;Operation;<op_name>;Schedule;

Imp;ProcessPlan;<process_name>;Operation;<op_name>;ResourcePool;Resource;

Imp;ProcessPlan;<process_name>;Operation;<op_name>;ResourcePool;Resource;CADModel;

Imp;ProcessPlan;<process_name>;Operation;<op_name>;ResourcePool;Resource;Personnel;

Imp;ProcessPlan;<process_name>;Operation;<op_name>;ResourcePool;Resource;Tools;

Imp;ProcessPlan;<process_name>;Operation;<op_name>;ResourcePool;Resource;Tools;Characteristic;

Example. The directive:

Imp;ProcessPlan;[a-zA-Z0-9_]*

Imports all process plan objects from the SAVE simulation request to CA.

Example. The directive:

```
Imp:ProcessPlan:Man[0-9]*
```

Imports all process plan objects whose name starts with the string Man followed by any digits.

2. Entries that require to Map the Type for process, material, and feature objects for CA. <-> is the separator between the left hand side and the right hand side. The entries on the left hand side correspond to the path in the SAVE data model to be traversed to get to target of the value to be imported. A valid path is any path from the msm Object onwards to a terminal attribute. So the path contains a set of tokens or keywords which are the actual names of the objects and attributes of the SAVE model sans the 'msm' prefix. The tokens or keywords are separated by semic-lons (;). The entries on the right hand side correspond to the CA nomenclature. The Process type, material type, and Feature type are required for CA.

The values for <ca_process_type>, <ca_material_type>, <ca_feature_type> have to exact names, or they can be the special symbol `*', in which case, the name from the SAVE object is used as is, i.e., the ProcessPlan name becomes the CA process type, the Material name for the Part/Material object maps to CA material type, and the Operation/OperationName or the Feature/FeatureName becomes the CA feature type.

```
Imp;ProcessPlan;<process_name>;RequiredType;<->;Process;<ca_process_type>;

Imp;ProcessPlan;<process_name>;Part;Feature;<feature_name>;RequiredType;<->;Feature;<ca_feature_type>;

Imp;ProcessPlan;<process_name>;Part;Material;<material_name>;RequiredType;<->;Material;<ca_material_type>;

Imp;ProcessPlan;<process_name>;Operation;<op_name>;RequiredType;<->;Feature;<ca_feature_type>;
```

Example. The entry:

```
Imp;ProcessPlan;[a-zA-Z0-9_]*;RequiredType;<->;Process;*;
```

Maps the name of the process plan to the type for Process object in CA.

Example. The entries:

```
Imp;ProcessPlan;Man[0-9]*;RequiredType;<->;Process;Manual;

Imp;ProcessPlan;Rob[0-9]*;RequiredType;<->;Process;Robotic;
```

Maps all process plan names which start with the string ‘Man’ and followed by any number of digits, to the CA process type ‘Manual,’ whereas all process plan names starting with ‘Rob’ are mapped to CA process type ‘Robotic.’

Example. The entry:

```
Imp;ProcessPlan;[a-zA-Z0-9_]*;Operation;[a-zA-Z0-9_]*;RequiredType;< >;Feature;Feature
```

Maps all operations to the CA top level feature of type ‘Feature.’

If, for example, the operations DRILL001, DRILL002, ... for a process plan DRILL are to map to CA feature ManDrill, but map to CA feature RobDrill if the process plan is ROBOTIC, the following entries are required:

```
Imp;ProcessPlan;DRILL;Operation;DRILL[0-9]*;RequiredType;<->;Feature;ManDrill;

Imp;ProcessPlan;ROBOTIC;Operation;DRILL[0-9]*;RequiredType;<->;Feature;RobDrill;
```

Note: *If there is more than one entry which matches, the mapping corresponding to the entry appearing last is used.*

3. All other entries that map to the CA process, material, and feature characteristics.<-> is the separator between the left hand side and the right hand side. The entries on the left hand side correspond to the path in the SAVE data model to be traversed to get to target of the value to be imported. A valid path is any path from the msm Object onwards to a terminal attribute. So the path contains a set of tokens or keywords which are the actual names of the objects and attributes of the SAVE model sans the 'msm' prefix. The tokens or keywords are separated by semicolons (;). The entries in the right hand side correspond to the CA nomenclature.

Example:

```
### Default CA Feature Char Name: FeatureName (text)

# Imp;ProcessPlan;<process_name>;Operation;<op_name>;Name;<->;Feature;<ca_feature_type>;Char;<ca_char_name>;

Imp;ProcessPlan;[a-zA-Z0-9]*;Operation;[a-zA-Z0-9_]*;Name;<->;Feature;[a-zA-Z0-9_]*;Char;*;
```

If the <ca_char_name> is ‘*’, the default name as specified in the comment above each entry in the template file is used.

The entries below show how the attribute ‘Name’ is mapped to a Feature Characteristic ManualDrillName or RoboticDrillName.

```
Imp;ProcessPlan;DRILL;Operation;DRILL[0-9]*;Name;<->;Feature;ManDrill;Char;ManualDrillName;

Imp;ProcessPlan;ROBOTIC;Operation;DRILL[0-9]*;Name;< >;Feature;RobDrill;Char;RoboticDrillName;
```

1.4 Specific Mapping for Some Operation Objects

The following describes how the Resource objects of an operation are mapped to Cost Advantage.

SAVE Attribute: Operation->PersonResApplic

This is a sequence.

Get the first object in the sequence of type `ResourcePool'

Get the Resource object

ResourcePool->Resource

The attributes for the Resource object are mapped to CA Feature characteristics.

### Default CA Feature Char Name: FeatureResourceName (text)

### Default CA Feature Char Name: FeatureResourceDescription (text)

### Default CA Feature Char Name: FeatureResourceEfficiency (numeric)

Get the Resource CAD Model object

ResourcePool->Resource->CadMod

The attributes for the CAD Model object are mapped to CA Feature characteristics.

### Default CA Feature Char Name: FeatureResourceCADModelName (text)

### Default CA Feature Char Name: FeatureResourceCADModelLocation (text)

Get the first object in the above sequence of type `Personnel'

The attributes for the Personnel object are mapped to CA Feature characteristics.

### Default CA Feature Char Name: FeaturePersonnelSkill (text)

### Default CA Feature Char Name: FeaturePersonnelLaborRate (numeric)

### Default CA Feature Char Name: FeaturePersonnelLaborRateYear (numeric)

SAVE Attribute: Operation->ToolResApplic

This is a sequence.

Get the first object in the sequence of type `Tool'

The attributes for the Tool object are mapped to CA Feature characteristics.

### Default CA Feature Char Name: FeatureToolsName (text)

### Default CA Feature Char Name: FeatureToolsDescription (text)

### Default CA Feature Char Name: FeatureToolsCost (numeric)

### Default CA Feature Char Name: FeatureToolsToleranceCap (numeric)

### Default CA Feature Char Name: FeatureToolsFailureRate (numeric)

### Default CA Feature Char Name: FeatureToolsType (text)

### Default CA Feature Char Name: FeatureToolsNumberOfBreakDowns (numeric, computed)

In case of importing attributes of msmConsumedParts and msmProducedParts, an index has to be specified to indicate the appropriate object from the list of objects.

1.5 Specific Mapping for Some ProcessPlan Objects

The following describes how the Resource objects of a process plan are mapped to CA.

SAVE attribute: ProcessPlan->PersonnelPool

This is a sequence.

Get the first object in the sequence of type `ResourcePool'

Get the Resource object

ResourcePool->Resource

The attributes for the Resource object are mapped to CA process characteristics.

### Default CA Process Char Name: ResourceName (text)

### Default CA Process Char Name: ResourceDescription (text)

### Default CA Process Char Name: ResourceEfficiency (numeric)

Get the Resource CAD Model object

ResourcePool->Resource->CadMod

The attributes for the CAD Model object are mapped to CA process characteristics.

### Default CA Process Char Name: ResourceCADModelName (text)

### Default CA Process Char Name: ResourceCADModelLocation (text)

Get the first object in the above sequence of type `Personnel'

### Default CA Process Char Name: PersonnelSkill (text)

### Default CA Process Char Name: PersonnelLaborRate (numeric)

### Default CA Process Char Name: PersonnelLaborRateYear (numeric)

SAVE Attribute: ProcessPlan->ToolPool

This is a sequence.

Get the first object in the sequence of type `Tool'

The attributes for the Tool object are mapped to CA process characteristics.

### Default CA Process Char Name: ToolsName (numeric)

### Default CA Process Char Name: ToolsDescription (numeric)

### Default CA Process Char Name: ToolsCost (numeric)

### Default CA Process Char Name: ToolsToleranceCap (numeric)

### Default CA Process Char Name: ToolsFailureRate (numeric)

### Default CA Process Char Name: ToolsType (text)

### Default CA Process Char Name: ToolsNumberOfBreakDowns (numeric, computed)

In case of importing attributes of msmMfgOrders, an index has to be specified to indicate the appropriate object from the list of objects.

Note: *Some hard-coded logic is used for importing attributes from msmSchedule. For a Schedule object in the SAVE db, the import client checks if any 2 of the 3 'Actual' attributes (ActualStartDate, ActualEndDate, ActualDurationHr) are set. If they are set then it gets those attributes based on the mapping provided for them in the import map file. If they ae not set then it gets the 'Planned' attributes (PlannedStartDate, PlannedEndDate, PlannedDurationHr) instead based on the mapping provided for them.*

2.0 The Export Client

This Client exports information from a Cost Advantage Note to the SAVE database. The binary executable is named client_out. The invocation of the client_out executable is as follows: client_out <hostname> <sim reqst.> <inputfile> <statusfile> <mapfile>

2.1 The Map File

The Map file for the export client can contain two types of entries:

- Filter entries
- Characteristic map entries

2.1.1 Filters entries

These entries indicate whether a particular object should be exported. These entries must conform to the following syntax:

Exp;Process;<regexp>;

OR

Exp;Feature;<regexp>;<parent-part-regexp>;

The Process filter will filter out any components/assemblies that do not match the filter specified. If no filters are specified, no assemblies or components will be exported. The Feature filter determines which of the features from CA should be exported. The first regular expression matches the feature type and the second matches the name of the part within which this feature is to be exported.

2.1.2 Characteristic maps

This section describes the syntax of the Export characteristic map entries. Each characteristic map entry provides information on the location of the value for particular SAVE attribute within the Cost Advantage Note. Each characteristic entry must appear on a single line. The syntax of an Export Map Entry is as below:

Exp;<LHS>;<-><RHS>;

The LHS represents a particular "path" to be traversed in the SAVE data model to get to the target of the value. A valid path is any path from the msm Object onwards to a terminal attribute. So the path contains a set of tokens or keywords which are the actual names of the objects and attributes of the SAVE model sans the 'msm' prefix. The tokens or keywords are separated by semicolons (;). Each ProcessPlan may obtain values only from within the CA Component that it corresponds i.e., values cannot be obtained from sub parts in case of assemblies. Using the hierarchy of SAVE objects given below some example <LHS> would be:

ProcessPlan;.*;AvgCriticalPath;

ProcessPlan;.*;Operation;.*;RefProcess;StdHrs

ProcessPlan;.*;Operation;.*;RefProcess;Characteristic;ref_id;TextValue

ProcessPlan;.*;Operation;.*;RefProcess;Characteristic;ref_id;NumericValue

Note: *Simulation Request is not included in the paths on LHS since it is under-stood that it is the starting point to reach any object in the SAVE data model.*

The <RHS> component of a characteristic map entry indicates the location of the value to be used for a particular terminal attribute within the SAVE model. The <RHS> may be any one of the following:

Exp;Process;<ftype>;Char;<cname>

Exp;Process;<ftype>;Cost;<cname>

Exp;Feature;<ftype>;Char;<cname>

Exp;Feature;<ftype>;Cost;<cname>

Exp;Material;<ftype>;Char;<cname>

Exp;Material;<ftype>;Cost;<cname>

Exp;TotalCost;Cost;<cname>

Note: *Assemblies do not have a material.*

2.2 Export Rules

Refer to the template map file map.export.template provided with the client software. It contains examples of mapping of all the attributes of the SAVE data model supported by CA. The template file needs to be copied as the file map.save.import within each cost model directory and edited as appropriate. The entries that have '#' in the first column are treated as comments.

The following broad rules are followed during the export:

1. A CA component or assembly maps to a SAVE msmProcessPlan Object.

2. Features in a CA component map to a SAVE msmFeature Object within the msmPart for the corresponding msmProcessPlan.

3. In the case of a CA Assembly, features map to an operation for the corresponding processplan.

4. Values within a CA component must be convertible to the expected target types. The client will attempt to convert these values intelligently wherever possible. For boolean targets, a not false value is assumed to be true (i.e. anything other than a 'false', 'no', or a 0 is assumed to be true).

5. In case of mapping a msmCharacteristic object, it is necessary to supply the name of the target characteristic. For instance if we desire to set the text value of a characteristic called Foo in the ProcessPlan's characteris-tics the map entry may look like:

Exp;ProcessPlan;.*;Characteristic;Foo;TextValue;<->;Process;.*;Char;Foo;

6. In case of mapping msmMfgOrders, msmConsumedParts and msmPro-ducedParts, an index has to be specified to indicate where the user wants to export the data in the list of objects. For example, if the user wants to export a CA Process characteristic called 'MfgOrderName' as the Name of the first msmMfgOrder object in the list of Mfg Orders in msmProcessPlan named 'PP1' then the map file entry would be:

Exp;ProcessPlan;PP1;MfgOrder;1;Name;< >;Process;.*;Char;MfgOrderName;

Appendix F

Factor AIM Wrapper User's Guide

Symix Systems, Inc.

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

1.0 Wrapper Overview

The purpose of the AIM wrapper is to integrate AIM with the architecture of Lockheed Martin's SAVE project. At this time, the wrapper is a functional part of the system at the level of the SAVE Phase III demonstration. This includes integration with the SAVE Data Model, Work Flow Manager, and other tools. This document describes the structure and use of the AIM wrapper, and outlines areas where assumptions have been made for purposes of the Phase III Demonstration.

2.0 Wrapper Architecture

The AIM wrapper is a separate process running on the same computer as AIM. It is a Windows application written in Microsoft Visual C++. The wrapper makes use of the CORBA interfaces defined for the SAVE project and implemented with Iona's Orbix environment. There is also an interface to the AIM database model, using ODBC.

2.1 Wrapper Functions

The AIM wrapper is activated by the receipt of a simulation request, either from the Work Flow Manager or through the wrapper's own user interface. The simulation request indicates which of the basic wrapper functions are to be executed. These are:

Flag Name	Action	Notes
Import	A new AIM alternative is created, by copying the baseline alternative and incorporating the new data from the SAVE Data Model.	The alternative to use as the baseline may also be specified. The default is alternative 000.
Edit	The selected alternative is opened for editing in the AIM Simulator.	These flags may be combined and are executed in the order shown. Since these functions work with a single alternative, if the Import flag was not used to create a new alternative, the operator's preference is used.
Simulate	The selected alternative is simulated in the AIM Batch Simulator.	
Gantt	The selected alternative's Gantt data is opened for viewing.	
Executive	The AIM Executive is opened, allowing the operator to perform any AIM actions desired.	This flag may not be combined with Edit, Simulate, or Gantt.
Export	The selected AIM alternative's input and output is exported to the SAVE Data Model.	If the alternative number is not clear from the previous actions (e.g. the Executive was used, which may have created more than one alternative), the operator's preference is used.

2.2 Wrapper Data Storage

When an AIM alternative is created (either through the wrapper's Import function or using the normal AIM functions), it is stored in its entirety in the AIM database, as usual. Alternative elements, which are unique to AIM, are not replicated in any other location.

When an AIM alternative is exported to the SAVE Data Model, the alternative data stored in the AIM database is translated into the schema of the Data Model and stored as attributes of new or existing objects. The corresponding AIM data remains in the AIM database. The SAVE Data Model may be used by other tools and possibly edited there. If the Data Model is then used to create a new AIM alternative, the wrapper creates a copy of the baseline alternative and updates it with components translated from the Data Model schema. The new alternative is a complete AIM model ready for editing and simulation.

3.0 Using the AIM Wrapper

The AIM wrapper requires AIM 8.x, Windows 95 or NT 4.0, and Orbix 2.3c or OrbixWeb 3.0. This section assumes that these components are in place and covers installation, configuration, and operation of the AIM wrapper.

3.1 Installing the Wrapper

The wrapper consists of two executable files. In addition, there is one AIM configuration file which is required to get AIM to support the full wrapper functionality. The wrapper is delivered in a Zip file containing these three files:

\aim\bin\AimWrap.exe

\aim\bin\batchmon.exe

\aim\projects\aimdbaux.dat

Note that the files must be installed in the appropriate subdirectories in the AIM directory structure.

3.2 Configuring the Wrapper

The AIM wrapper accesses AIM databases as ODBC data sources. You must create one or more ODBC data sources to tell the wrapper where to find the AIM databases. To do this, go to the Control Panel and open the ODBC icon (named "32-bit ODBC" in Windows 95). On the "User DSN" tab, click on Add to define a new data source. The only requirements are to give the data source a unique name, and select the AIM database (an .MDB file, probably in the \aim\projects directory) associated with this data source. Then click OK to save the data source definition. More than one data source can be created if you want to use more than one AIM database.

Also, since the AIM wrapper acts as an Orbix server for the Work Flow Manager, it must be registered with the Orbix installation on the machine where it runs. To do this, run the following command from the Start menu's Run dialog or from an MS-DOS prompt:

```
putit  AimSimulator  c:\aim\bin\AimWrap.exe
```

Be sure to replace the c: with the drive where the AIM Wrapper is actually installed.

In addition, Orbix may require that the IP address of the computer running the SAVE database server be added to the "hosts" or "lmhosts" file on the computer running the wrapper.

3.3 Running the Wrapper

After the wrapper has been installed and configured, you can start it by running the executable file \aim\bin\AimWrap.exe. AIM should not be running at the time. There are two command line arguments to AimWrap.exe:

1. name or IP address of the computer running the SAVE database server

2. name of the ODBC data source associated with the AIM database you want to use

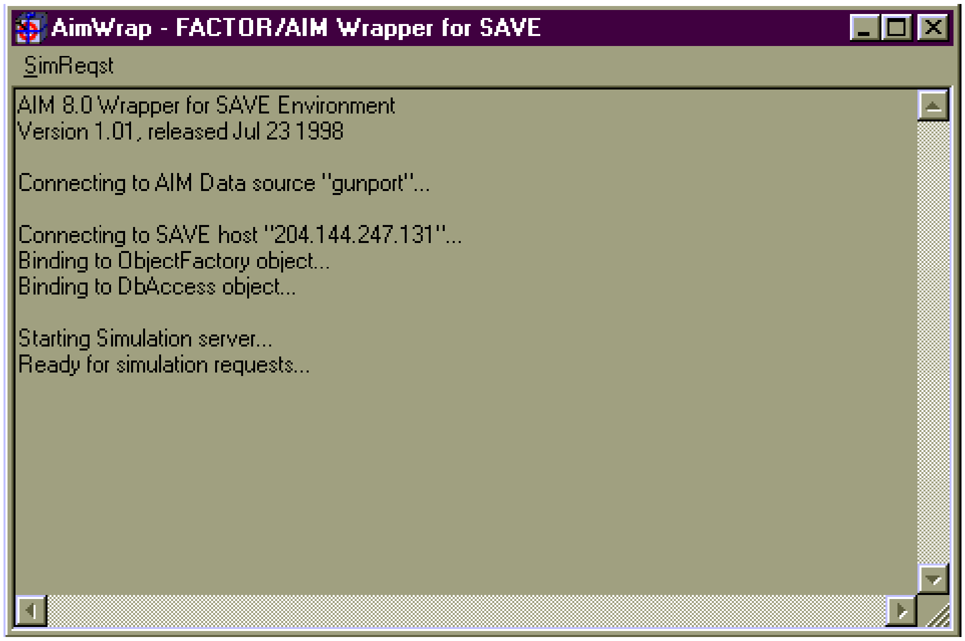
Both arguments must be entered each time the wrapper is run. If you will be running the wrapper repeatedly with the same SAVE server and ODBC data source, it will be useful to create a shortcut on the desktop or under the Start menu, with the arguments entered on the shortcut's command line.

For both arguments, enclose the name in quotes if the name includes a space. For example:

```
AimWrap  servername "ODBC data source name"
```

3.4 Using the Wrapper

When you start the wrapper, you should see a window like this:



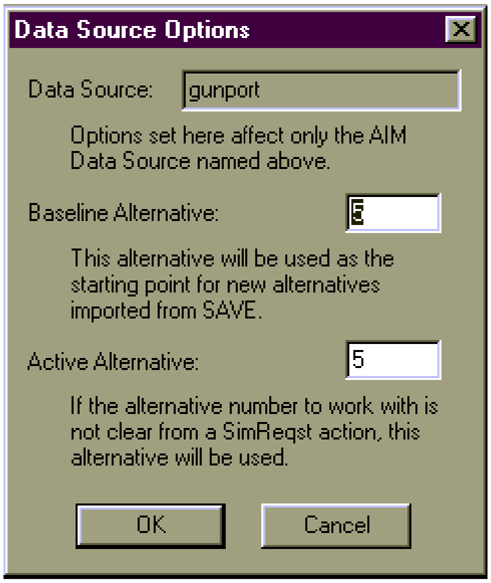
```
AIM 8.0 Wrapper for SAVE Environment
Version 1.01, released Jul 23 1998

Connecting to AIM Data source "gunport"...

Connecting to SAVE host "204.144.247.131"...
Binding to ObjectFactory object...
Binding to DbAccess object...

Starting Simulation server...
Ready for simulation requests...
```

The center of the window presents status information on the wrapper session. The menu bar and the standard window icons in the title bar allow you to manipulate the wrapper window.

The following sections outline the functions you can perform with the wrapper.

3.4.1 Setting preferences for the current ODBC data source.

To do this, select the "Data Source Options" item from the SimReqst menu. You will see the dialog shown at right.



Data Source: gunport

Options set here affect only the AIM Data Source named above.

Baseline Alternative: 5

This alternative will be used as the starting point for new alternatives imported from SAVE.

Active Alternative: 5

If the alternative number to work with is not clear from a SimReqst action, this alternative will be used.

The "Data Source" box shows the name of the current data source. This is the data source that was entered on the AimWrap.exe command line. Options set in this dialog affect only this database.

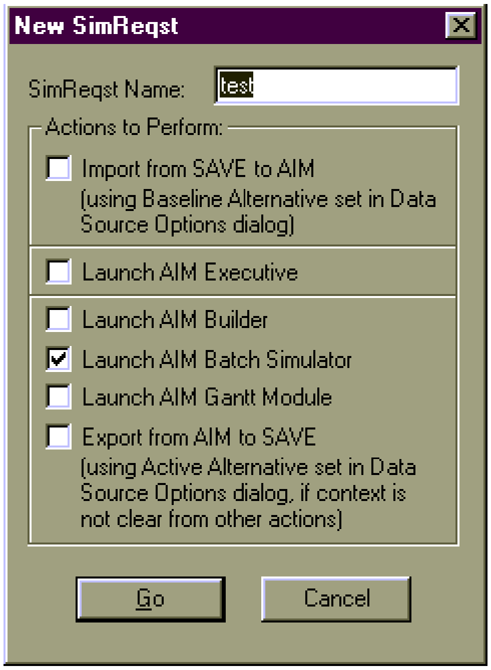
The "Baseline Alternative" box allows you to select which AIM alternative to use as the starting point when creating new alternatives in this database.

The "Active Alternative" box allows you to select which AIM alternative to use in situations when the alternative number is not clear from a simulation request action.

Note: the options you set in this dialog are saved in a file in the directory specified by the TEMP environment variable. The file will have an .INI extension, and its name will be the same as the data source name. For example, if TEMP is set to "c:\temp" and the data source name is "gunport", the information will be saved in file c:\temp\gunport.ini. You may delete this file if you want to restore the default settings for the data source, or when you are finished using the data source with the AIM wrapper.

3.4.2 Initiating a new simulation action.

The AIM wrapper allows you to manually start a simulation request action. To do this, select the "New SimReqst" item from the SimReqst menu. You will see this dialog:



New SimReqst

SimReqst Name: test

Actions to Perform:

- □ Import from SAVE to AIM (using Baseline Alternative set in Data Source Options dialog)
- □ Launch AIM Executive
- □ Launch AIM Builder
- ☑ Launch AIM Batch Simulator
- □ Launch AIM Gantt Module
- □ Export from AIM to SAVE (using Active Alternative set in Data Source Options dialog, if context is not clear from other actions)

The "SimReqst Name" box allows you to enter the name of a SimReqst object to use. If an object with this name does not exist, the AIM wrapper will create one.

The "Actions to Perform" box allows you to select which of the wrapper functions you want to include in this simulation request. These actions correspond to the basic wrapper functions defined in section 0.

When you click on the "Go" button, the simulation request will be created and the wrapper will begin executing the selected functions.

Note: the settings you enter on this screen are saved in a file named aimwrap.ini in the directory specified by the TEMP environment variable. They will be used as the default for the next time you enter the dialog. You may delete this file if you want to restore the default settings, or when you will no longer be using the AIM wrapper.

3.4.3 Responding to Work Flow Manager requests.

In addition to manual operations, the AIM wrapper will also react to messages from the Work Flow Manager. No user intervention is required for this function. As long as the wrapper is running, it is ready to respond to Work Flow Manager requests. When a request is made, the status of the action will be shown in the main section of the wrapper window.

3.5 Closing the Wrapper

When your wrapper session is complete, you can close the window with the SimReqst/Exit menu item, or with the standard window controls in the title bar. As you close the window, the wrapper will terminate its connections and close all open files.

Note: as it runs, the wrapper creates a file named aimwrap.txt in the directory specified by the TEMP environment variable. This file contains a log of all the status messages sent to the wrapper window during the most recent wrapper session. This file may be useful for debugging or documenting the wrapper's behavior. It can be deleted at any time after the wrapper is closed.

4.0 Limitations

The AIM wrapper developed for the SAVE Phase III Demonstration has limited support for some of the features of AIM. This section lists the limitations known at this time.

4.1 Problem Definition Limitations

- The AIM wrapper expects summary output data to be available once the AIM simulation has been completed. In general, all summary flags in the AIM problem definition (visible under the Project/Problem menu item in the AIM executive) should be enabled.

- The AIM wrapper also makes use of order performance and Gantt data to publish the schedule information to the SAVE Data Model. Therefore, the Order Performance flag and Gantt flag in the AIM problem definition should both be enabled.

4.2 General Modeling Limitations

- If model components are deleted or renamed in either SAVE or AIM, the change is not reflected in the transfer.

- An AIM alternative should contain exactly one main process plan and at least one sub-plan. (Sub-plans are modeled with Batch jobsteps; see below).

- An AIM alternative should contain at least one demand order. This is used to set the SAVE model's build rate.

- An AIM alternative should contain at least one resource group, one resource, and one material.

4.3 Process Plan Limitations

- AIM process plans may include any type of jobstep. The AIM wrapper imports and exports full jobstep functionality for the following jobstep types:

  - Setup/Operate

  - Operate

  - Assemble

  - Setup

  - Move

  - Batch

  - Add-to-Material

  - Remove-from-Material

- Jobstep types not listed above will be exported to the SAVE Data Model as if they were Operate jobsteps.

- Nested process plans are implemented by using a Batch jobstep on the main process plan. The batch definition should be a remote batch, with its own unique process plan. The process plan should have the same name as the batch definition.

- Consumed parts are implemented in AIM by using an Assemble or Remove-from-Material jobstep. Each part is associated with an AIM material.

- Produced parts are implemented in AIM by using an Add-to-Material jobstep. Each part is associated with an AIM material.

- Resources allocated on a jobstep should always be resource groups, not simple resources.

- Jobstep step times should be simple floating point numbers. The jobstep time rule should be "Total jobstep time".

- When moving data between AIM and SAVE, jobsteps should not be re-ordered, renamed, or deleted.

4.4 Resource Group Limitations

- Resource groups may only contain identical resources.

- Since the SAVE Data Model does not maintain information on group members, the wrapper expects that members of group GROUP1 be named GROUP1.1, GROUP1.2, etc.

- Resource groups should not contain more than 20 members.

4.5 Shift Limitations

- Operator resources should be associated with only one shift. (Machine and other resources will not have shift information exported to or imported from the SAVE Data Model.)

- Since shifts in the SAVE Data Model refer to single-day normal working hours, AIM shifts will be exported to SAVE as starting at the earliest start time of any day of the week, and ending at the latest end time of any day of the week.

- When importing SAVE shift data to AIM, existing AIM shifts will not be updated with new time data from SAVE. New shifts will be created to span Monday through Friday with the start and end times specified on the SAVE shift.

- The SAVE break component is not used by the AIM wrapper.

4.6 Work Schedule Limitations

- The AIM Work Schedule component is effective for all resources and shifts. When exported to the SAVE Data Model, all personnel will be linked to the WorkCalendar. When imported from SAVE, all WorkCalendar records will be imported, but any connection to specific personnel will be lost.

4.7 Order Limitations

- Demand order ID should be the same as the ID of the part being ordered.

- Demand order time between arrivals should be a simple floating point number. This value (in hours) is converted to a monthly build rate based on 30 days per month.

- Load size should always be 1, so that material requirements on Assemble jobsteps will be consistent with part flow through the process plan in the SAVE Data Model.

4.8 Simulation Output Limitations

- The output of the simulation exported to the SAVE Data Model includes the schedule for each demand order (exported to SAVE as a MfgOrder) and the overall schedule for the main process plan.

Appendix G

IGRIP and QUEST Wrapper User's Guides

Deneb Robotics

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

1.0 Startup Commands

1. Start the SAVE data base server

2. Start the Work Flow Manager

3. Start a shell in the Unix workstation, use the ping command to verify that the Unix workstation can talk to the WFM PC and the Server PC using the names exactly as in the "save.cfg" file.

4. Verify that Orbixd is running, [ps -ef | grep orbixd]

5. Start a new shell and from the /usr/deneb/save directory start the IGRIP WFM simulator server using the command "`simulator igripsim`"

Note: These are the same names as in the "save.cfg" file. If you decide to use different names here then you have to modify the names in the "save.cfg" file also.

You will see a screen as shown below:

```
Diagnostics level = 0

STARTING NON-BLOCKING SIMULATOR SERVER

Starting SIMULATOR CORBA server ...SIMULATOR CORBA: Support initialized

----------------------------------------------------------------------

COMMANDS:

        exit    Exit server

        send    Send status information back to listener

        status  Print server status information


TYPE COMMAND AND PRESS ENTER TO ACTIVATE IT!
```

6. Open another shell and Launch IGRIP from /usr/deneb/vmap by typing "igrip". See user instructions in the next section to verify connectivity with the Server and WFM.

7. Start a new shell and from the /usr/deneb/save directory start the QUEST WFM simulator server using the command "`simulator questsim`". Again you will see a screen as above.

8. Open another shell and Launch quest from /usr/deneb/save by typing "save_quest". See user instructions in the next section to verify connectivity with the Server and WFM.

9. Use work Flow Manager to start a process, and launch IGRIP, refer to WFM instructions on how this is to be accomplished.   You will see

10. Use work Flow Manager to start the same process and launch QUEST

2.0 User Notes

Assumptions:

1. User is familiar with general IGRIP and QUEST use and knows the basic menu system navigation and terminology.

2. WFM, SAVE server, IGRIP & QUEST are up and running.

3. SAVE server has some valid data and the name of at least one SimRequest in the SAVE database is known.

2.1 General Information

Reading and writing information to the SAVE server database is achieved by selecting buttons on the User pages. Several user pages have been set up for IGRIP and QUEST.

A picture of the three user pages used is shown in the next page.

Many of the buttons are the same between IGRIP and QUEST and the underlying functions behave as appropriate to IGRIP or QUEST.

Once connection is established with the SAVE server, there is an always an active SimRequest, within the SimRequest there is an active Process Plan.

The active objects can be changed to point to some other object of similar type as required. The interaction with the database is always live, in the sense that after every change and conformation of the change the data is immediately written to the Database. If you modify parameters of an Operation it is immediately written to the Database. We do not read in the entire Process plan and then write all the operations back when only one operations has been modified. We read and write back as requested.

2.1.1 IGRIP

IGRIP workcell simulation data can be thought of as a starting point to input data into the SAVE server to be used by other software tools. Generally speaking the user creates a workcell model and creates a simulation of a particular Operation or from a small nested process plan. The total simulation time for an operation can then be written to the database. Once a simulation has been run the cycle time for a particular device can be obtained from IGRIP device properties and then written to a particular operation. A workcell model is not built from the data in the Save database. While Process plans & Operations can be created from IGRIP and a portion of the data for each of these objects can be entered, these features are provided so the user does not have to use the Query Manager for simple changes.

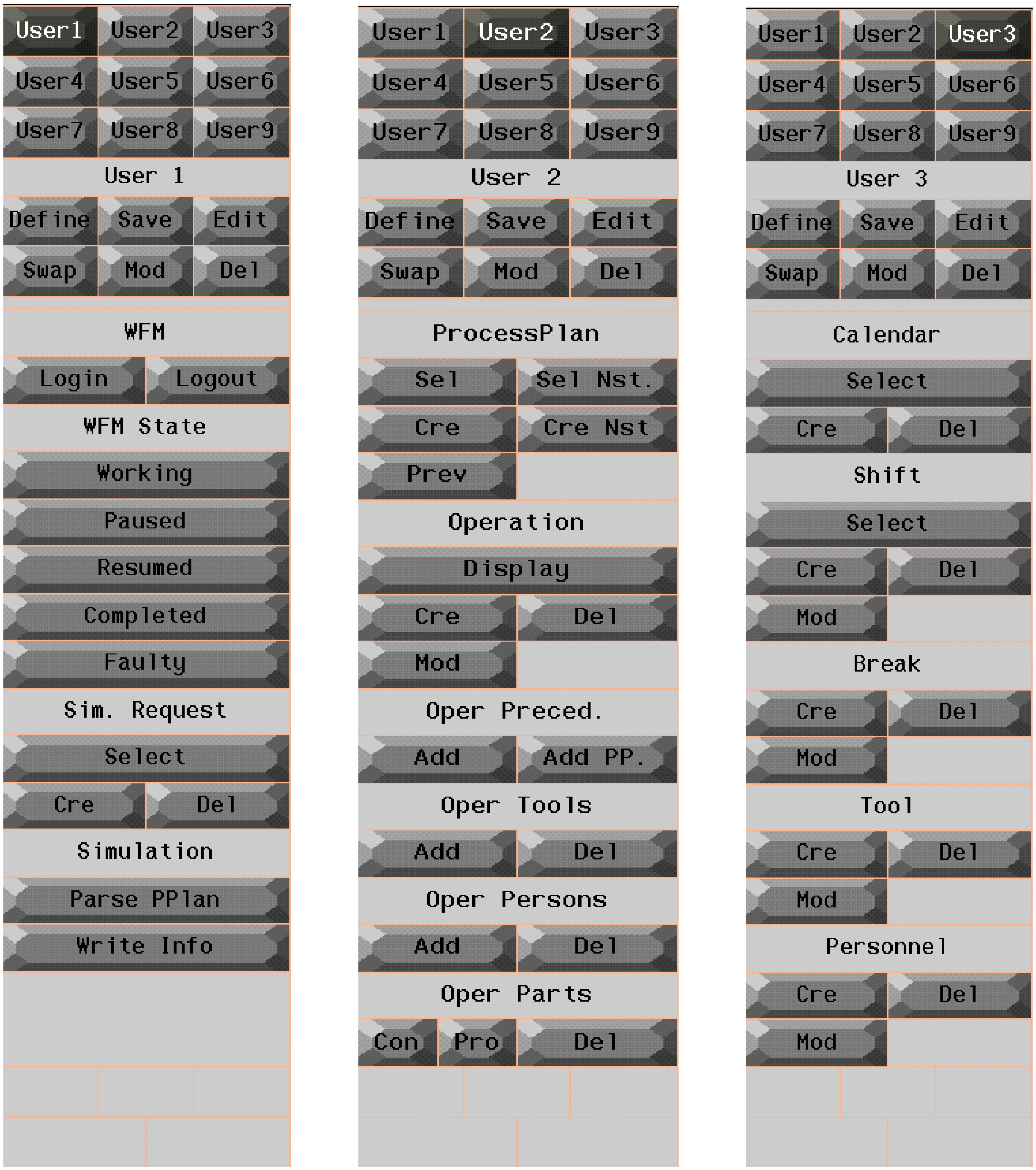


Figure G-1: User Interface Buttons

2.1.2 QUEST

From a particular process plan in the SAVE server an entire QUEST model can be built, of course this depends on the completeness of the information in the Process plan. For example the resources and labor are created automatically with the default geometry unless the complete path name of the geometry to be used are also specified in the SAVE model. Once a process plan from which a QUEST model is to be built is selected the entire Process Plan including all the nested process plans and operations are read into the memory of the QUEST process plan parser program. The memory model is then parsed in 4 passes and information required for QUEST is computed. A list of commands in QUEST BCL format are then created and sent to QUEST for model creation. These commands can also be written to a BCL file. A model template can also be specified as a starting point as well as several configuration files for user BCL commands. The names of the files are specified in the "save.cfg" file. The purpose and description of each file follows. These files can be used to customize the QUEST model after it is created from the SAVE database.

QUEST_BCL_FILE

BCL command file, which is created when process plan is parsed.

QUEST_ORDER_FILE

Order file, which is created by the parser program from the database. The QUEST model during simulation reads the same file.

QUEST_SCL_TEMPLATE

The scl-logic file, what is used in the parsed simulation model.

QUEST_SCL_FUNCTION

The function name in the "QUEST_SCL_TEMPLATE" file, which is used as the default source logic in the QUEST model. This logic is automatically assigned to the QUEST model source resources.

QUEST_TEMPLATE_BEGIN

BCL template file name. This file may contain user defined BCL commands, which are appended to the beginning of all parsed BCL commands.

QUEST_TEMPLATE_END

BCL template file name. This file may contain user defined BCL commands, which are appended to the end of all parsed BCL commands.

QUEST_BCL_DIRECT

If this flag is set to "1", then QUEST will execute parsed BCL commands directly (BCL commands are still written into the file).

AUTOPOSITIONING

If this flag is set to "1", then parsed QUEST model will use a grid to automatically position all machines.

AUTOPOSITION_GRID

Auto-positioning grid distance in mm.

DEFAULT_TOOL_GEO

Use this filename from ..lib/DEFAULTS To display tool default geometry.

RESERVED_WORDS

These words cannot be used in QUEST as part, process, machine, labor, tool names

PPLAN_HEADER_LENGTH

If this value (integer) is other than Zero, then system truncates processplan name to use equivalent amount of characters.

QUEST_PROCESS_NAME

This defines, what field in the operation (database) is used in the QUEST model to describe QUEST process name. This field can use following values:

id	Operation id
name	Operation name
description	Operation description

The default value is "name"

RESOURCE_FILE

Temporary file to store QUEST resources

GEOMETRY_LOC

0 = use matrix, 1 = use CADmodel envelope (fake)

GEOMETRY_NAME

0 = inquire actual file name, 1 = use part name

GEOMETRY_CADMOD

0 = Use default geom, 1 = use CadMod geometry

SCHEDULES_DIRECTORY

Directory where schedules files (specifying calendars) will be saved.

OBJECT_REF_MODE

What constitutes a way to uniquely identify operation objects. This is important for the parser because it construct links between object, so it needs a way to identify the object before linking another object to it.

0 = (id:name:description) (default)

1 = name

2 = stringified CORBA reference

2.2 Establishing Connection to the SAVE Server

These instructions apply to IGRIP and QUEST

2.2.1 Selecting the SimRequest

Before you can perform any SAVE action, you must select the SimRequest you want to use:

USER | PAGE1 | Sim. Request | Select

(This is the normal Deneb methodology of representing the button to be activated or selected.)

This means select the "User" Context (the top row of buttons), then select "Page1" at the top right hand corner from the set of nine buttons called the Page Buttons. This will result in several buttons displayed on the right hand side called the action buttons. Look for the title "SimRequest" and then activate the button called "Select". The button will turn purple to indicate it has been selected.

This action displays a dialog box where you can specify a server by workstation network name and simulation request name. The server name and simulation request names are initialized from "save.cfg" configuration file but can be changed here if needed.

After you accept values in the dialog box, by selecting "Done" in the dialog box a connection is established to the SAVE server and the specified SimRequest.

If something fails, then use the error message to verify:

a) The server (name) exists and you are able to connect to it. Use Ping or other tools to verify connectivity.

b) Make sure that SAVE server program is running in the server host.

c) Make sure that you have the correct permissions to use the SAVE server

When connection and simulation request is selected correctly, you are able to use other functions.

DISTRIBUTION STATEMENT C: Distribution authorized for U.S. Government Agencies and their contractors. Other requests for this document shall be referred to the PEO JSF.

2.2.2 Selecting a Process plan

The next step is to select a Process plan to be the active Process Plan.

When simulation request is selected, it automatically selects the first Process Plan object as the current Process Plan.

You can change the current Process Plan to point to the Design Alternative Process Plan list by selecting USER | PAGE2| ProcessPlan | Sel

A list of process plans that are attached to the current SimRequest is displayed.

Select one of the process plans from the list to be set as the active Process Plan.

NOTE: The top most selection is the SimulationRequest->ProcessPlan Object, which is used as a default. All other Process plans displayed in the list are Design Alternative Process plans.

2.2.3 Displaying Operation information

User can display operation information by selecting

USER | PAGE2 | Operation | Display

A list of operations that are part of the current Process Plan is displayed.

From the Operation dialog box select the Operation for which additional information is to be displayed. Another dialog box with the properties of the Operation is displayed. This cycle can be repeated until user "aborts" operation selection.

At this stage the connection to the server is established and the Data from the server can be read from IGRIP or QUEST as the case may be.

2.2.4 Disconnecting from the server

USER | PAGE1 | Sim. Request | Logout

To ensure that the last transaction with the Database has been completed it is required that the user use the Logout function to disconnect properly from the server. This also ensures that all the temporary objects created by the server are also deleted.

2.3 Establishing Connection to the Work Flow Manager

These instructions apply to IGRIP and QUEST, Refer to Component block diagram on Page **Error! Bookmark not defined.**. A connection to the SAVE server can be established regardless of the connection with the WFM.

When the SIMULATOR server was started it would have already established a connection with the Work Flow manager using the configuration parameters from "save.cfg" file. Once QUEST or IGRIP is started then the user can then establish a connection between QUEST or IGRIP and the SIMULATOR server using USER | PAGE1 | WFM | Login. This informs the simulator server that QUEST or IGRIP is being used by a user and will allow any messages sent from the WFM to be displayed in the Message window. The buttons "Working", "Paused", "Resumed", "Completed", "Faulty" is to be used to interact with the WFM based on the message sent from the WFM. The user must use the Logout function prior to exiting from IGRIP or QUEST. User has to logout from WFM as well as the SAVE server. These are two independent actions.

2.4 User Function DescriptionsCommon to IGRIP and QUEST

2.4.1 Create New Process plan

User can create a new process plan within the Designalternative list by selecting

USER | PAGE2 | Process Plan | Cre

User has to type the process plan name and description into the dialog box and accept the process plan creation by selecting "Done".

NOTE: If you want to set this newly created process plan as "active", you must use "Process plan | Sel" function.

2.4.2 Select Nested Process Plans

To select a Process Plan that is nested within an Operation as the active plan use USER | PAGE2 | Process Plan | Sel Nst.

A list of nested process plans that are attached to the current Operation is displayed. Select one of the process plans from the list to be set as the active Process Plan.

2.4.3 Create Nested Process Plans

Not yet implemented

2.4.4 Selection of Parent Process Plan

Not yet implemented

USER | PAGE2 | Process Plan | Prev

Will add the capability to go back one step to the parent process plan from a nested process plan, otherwise you have start at the top from the SimRequest.

2.4.5 Create Operation

User can create a new operation in the active process plan by selecting

USER | PAGE2 | Operation | Cre

First user must add some basic information (name, description, id, runtime, setup time) in the dialog box. The user also enters in the dialog box whether the operation includes a nested process plan by answering yes or no to the “Process Plan Include” field. If the Operation includes a process plan, the user can specify whether to expand or collapse the nested process plan, by choosing and entering either “expand” or “collapse” to the filed “Expand Plan”. After dialog box values are accepted, system will create a new operation. After operation has been created, system will ask user to select "Precedent" operations. This action will "loop" until user selects abort, after which system will update these precedent operations into the newly created operation.

Note: Only operation within the current Process Plan can be selected as precedent operations.

2.4.6 Delete Operation

Not yet implemented

2.4.7 Create Part Information for Operation

User can create operation consumed part and produced part information by selecting

USER | PAGE2 | Oper Parts | Con for consumed parts

USER | PAGE2 | Oper Parts | Prod for consumed parts

User has to select an operation from the list and then select a Part from a Model or Workcell. If the Part already exists, then user has to select, whether to create a new Part or to overwrite the existing part. The user cannot enter quantity or part location fields for consumed parts and produced parts. The user cannot modify part usage information.

The geometry of the part can be specified by giving the full path name of the file containing the CAD data, under “CAD data file:” prompt, or can be specified interactively be entering “Yes” to “Interactive Select” prompt. In this case, the user will be given a list of CAD files to choose from. In either case, the full path name of the CAD file is stored in the database. If the user enters “No” to “Interactive Select:” and leaves the “CAD file:” prompt blank, then the default part geometry is used. The above also applies to selection of resource CAD geometries.

2.4.8 Delete Part Information for Operation

Not yet implemented

2.4.9 Write Simulation Information

User can write simulation information by selecting

USER | PAGE1 | Simulation |Write Info

The SimMod object attached to the SimRequest field is modified. A dialog box is provided with several fields. The specific fields are Name of simulation model, Description, Configuration file name, etc. After user enters the information it is written into the database.

2.5 IGRIP SPECIFIC FUNCTIONALITY

2.5.1 Modify Operation information

The user can retrieve IGRIP devices cycle time interactively from the existing workcell. And the modify operation Runtime information by selecting

USER | PAGE 2 | Operation | Mod

First user must select the Device, for which the cycle time is to be used as a reference for the operation. After user has successfully selected the Device, the user must then select the related operation. When operation has been selected successfully, system will display a dialog box, which displays first old values and then new values. User can edit these new values if required and then accept or abort updating. If user accepts values, then they will be written to the database.

Note: The "INTERACTIVE_UPDATE" flag in the save.cfg configuration file must be set to 1 for this functionality.

2.6 QUEST SPECIFIC FUNCTIONALITY

2.6.1 Create QUEST model from database (Parse process plan)

User can create a QUEST model by selecting a process plan by using

USER | PAGE1 | Simulation | Parse PPlan

After successful process plan selection the system starts to parse this process plan in order to create all required BCL commands to build a QUEST model. This operation will take some time depending on the size of the process plan. To show that the data is being read from the server you will see dots (.) appearing in the message window. Each dot represents an operation in the process plan. All operations, nested process plan operations are also read. Once the data is read into memory all QUEST required information is computed and then interaction with QUEST starts to create the QUEST model. For a 170 step process plan this operation can take as much as 4 minutes, depending on the speed of the SGI workstation as well as the server speed.

NOTE:

User can also define a BCL template, which will be appended in the end of all parsed BCL commands. This template can be added to do specific visualization, modeling or simulation tasks.

If "QUEST_BCL_DIRECT" flag is set to "1" in the save.cfg, then parsing will send all commands also directly into the QUEST.

2.6.2 Modify Operation

To modify the values of any field of a particular operation use the function

USER | PAGE 2 | Operation | Mod. The user has to select an Operation from the list of Operation for the Process Plan.

2.6.3 Add precedent Operations

USER | PAGE2 | Oper Preced. | Add

After an operation is created this function can be used to add precedent Operation(s). This action will "loop" until user selects abort, after which system will update these precedent operations into the newly created operation.

Note: Only operations within the current Process Plan can be selected as precedent operations.

2.6.4 Add Precedent Process Plans

After an operation is created, the user can add precedent Operation(s) that belong to some other parent process plan by using the function USER | PAGE2 | Oper Preced. | Add PP. This action will "loop" until user selects abort, after which system will update these precedent operations into the newly created operation.

2.6.5 Add tool(s) to an operation

User can add tools to an operation by selecting

USER | PAGE2 | Oper Tools | Add

First user must select the operation, where tools need to be added. Then user can add tools in the "Select Tool" dialog box. From this dialog box user can change what tool to use and specify how many tools an operation requires (User must not use more tools than is available). In the "Select Tool" dialog box it is also possible to define the tool location in the QUEST model.

In your save.cfg configuration file, there is an option called

"Autopositioning". If it is specified as "No", QUEST will use the XYZ location

stored in the database to position the resources. If the user did not

enter any values, and autopositioning is "NO", QUEST will think that those values are 0,0,0 and all resources will appear in the middle.

If you specify autopositioning as "Yes", then you should also specify the

"Autopositioning grid size (in mm)" in your configuration file, and QUEST will line up the resources throughout the factory (ie put them side by side in a line, separated by a distance equal to the autopositioning grid size and go on to the next line when it runs out of space).

NOTE:

One and only one tool application, where the tool is of type = ‘stationary’, must be assigned to every Operation.

Process plan must contain at least one tool pool, before tools can be added into the operation (USER | PAGE3 | Tools | Cre).

2.6.6 Delete tools from an operation

Not yet implemented

2.6.7 Add person(s) to an operation

User can add persons into the selected operation be selecting

USER | PAGE2 | Oper Persons | Add

First user must select the operation, where persons need to be added. Then user can add persons in the "Select Person" dialog box. From this dialog box user can change what person to use and specify how many persons an operation requires (User must not use more persons than available).

NOTE: Processplan must contain at least one personpool, before person(s) can be added into the operation (USER | PAGE3 | Personnel | Cre to add a personpool).

2.6.8 Delete persons from an operation

Not yet implemented

2.6.9 Calendars

The user can create a work calendar object into the database by choosing:

USER | PAGE3 | Calendar | Create

The user will need to enter a name, description, year, and day of week Jan 1st falls on. Once created, a calendar can then be assigned to certain personnel elements.

To select a calendar to be the active calendar select

USER | PAGE3 | Calendar | Select.

A list of calendar will be displayed and the user is able to select a particular calendar as the active shift for U/I purpose only.

2.6.10 Shifts

To create a shift select USER | PAGE3 | Shifts | Create. The User will be required to enter Start Time and End time in number of hours since midnight. Decimal portion in tenths, ie: 0.5 = 30 minutes.

To select a shift to be the active shift select USER | PAGE3 | Shifts | Select. A list of shifts will be displayed and the user is able to select a particular shift as the active shift for U/I purpose only.

To modify the parameters of a shift select USER | PAGE3 | Shifts | Modify.

2.6.11 Break

To add a break into the active shift select USER | PAGE3 | Break | Create.

The User will be required to enter Start Time and End time in number of hours since midnight to specify the break as well as a name. Decimal portion in tenths, ie: 0.5 = 30 minutes.

A specific break can be selected and the fields modified by selecting USER | PAGE3 | Break | Modify

2.6.12 Create tool pool (machines)

User can create new tool pool (Machines) into the selected process plan by selecting

USER | PAGE3 |Tool | Cre

In the "Type Tool Information" dialog box user must type tool Name, Description, Quantity of tools available, and Type. For Type, the user must select between the two choices given, which are “stationary” or “movebale”.

2.6.13 Modify tools pool

User can modify tool pools by selecting

USER | PAGE3 |Tool |Mod

In the "Edit Tool Information" dialog box user can edit tool name, description and Quantity of tools available.

2.6.14 Create Personnel

User can create new persons into the selected process plan by selecting

USER | PAGE3 |Personnel | Cre

In the "Type Person Information" dialog box user must type person name, description, Quantity of persons available and Efficiency.

2.6.15 Modify personnel

User can modify persons (machines) by selecting

USER | PAGE3 | Persons | Mod

In the "Edit Person Information" dialog box user can edit person name, description, Quantity of persons available and Efficiency.

2.6.16 Updating Utilization information.

USER | PAGE4 | Update Util | Person Util

USER | PAGE4 | Update Util | Tool Util

User can update utilization information for either tool or personnel, in either case system will ask for tool or personnel name. The user should enter the name as it is in the SAVE database. Then the system will ask the user to enter the (QUEST) resource name. These names could be different in case the SAVE database name is a QUEST key word or duplicate name. Usually a number would have been appended to the name. The QUEST calculated utilization statistics from the simulation run is written into the database.

2.7 General Additional User Information

2.7.1 Display of lists in interface

The SAVE environment involves the display of lists. For example, list of process plans that exist in simRequest, list of operations that exist in process plan, list of tools that exist in operation. When a list consists of two elements, *the GUI will only show one element*. You have to click on that element to see the second element to select it. Therefore, when you add an element to a single element list, it may look like it did not work when in fact it did.

2.7.2 Display of Operation

When displaying an operation, all data pertaining to that operation is displayed EXCEPT for the number of units under each tool and each personnel.

2.7.3 Displaying Tools, Personnel, Calendars and Shifts available to a process plan

Currently, there is no direct way of displaying them. To see the list of tools or personnel, you have to try to add a Tool or Personnel to an existing operation. The GUI will then show you a list to choose from.

To see the list of calendars and shifts, you have to try to associate a calendar or a shift to existing personnel. The GUI will then show you a list to choose from.

2.7.4 Precedent Operations

Precedent operations must be regular operations, not nested process plans. If a user sets a nested process plan operation to be the precedent of another operation, it is allowed, but then the process plan parser will change it so that *the last operation* in the nested process plan is the precedent operation. Here, *last operation* means operation entered last by the user, which may or may not be the *logical* last operation.

2.7.5 Adding Tools or Personnel to Operations

When adding tools or personnel to operations, the GUI will display all tools and personnel in all process plans, not only in the active process plan.

2.7.6 Production rate

We currently do not use or take into account production rate or manufacturing order information (msmMfgProgram or msmMfgOrder objects in database). This can be done in the future.

2.7.7 Resource Usage

If an Operation requests to use more units of a resource (labor or tool) than is specified in the resource pool size, the system will just hang waiting for the resources to be available, which they will never be. So it is up to you to make sure this does not happen.

2.7.8 CAD File Path Name

Regarding specification of the full path name of the CAD files when the file is on the network: If the network drive has been mounted properly, then the directory can be treated as if it were local. IE just specify The Full Path Name From Root.

3.0 Summary of Changes For Final Demo

3.1 Tool Type

In this final phase implementation, an assumption regarding the tool types in an msmOperation's tool application list was removed. The assumption was that the first tool in the tool application list is a "stationary" tool and therefore corresponds to a Quest "machine". In this final implementation, the attribute "type" associated with a tool contains either the string "stationary" or the string "moveable" which indicates whether it will be a Quest machine or a Quest moveable tool (internally, a labor). Therefore, each operations tool application list must contain **one and only one** tool application with the tool of type stationary.

3.2 Nested Process Plans

In the SAVE database model, an msmOperation can point to a process plan. This process plan is nested within that operation. In the interim demo presentation, the Quest parser automatically "exploded" the operation into its nested process plan. In this final demo implementation, a characteristic within operation, with the name "QUESTflag", holds a numerical value of either 0.0 or 1.0. When the Quest interface is used to specify the QUESTflag characteristic, instead of entering numerical values, the user is prompted to select between two options "expand" and "collapse" under a new "Expand Plan?" field when creating an operation. Internally, "expand" corresponds to 1.0 and collapse to 0.0. If the value is 1.0, the parser will explode that operation. If it is 0.0, it will not explode it but *will treat it as a simple operation*. If a characteristic with the name "QUESTflag" cannot be found, the parser will explode the operation.

3.3 Consumed Parts and Produced Parts

An msmOperation has a list of consumed parts and a list of produced parts. These are actually lists of msmPartUsage objects, where a msmPartUsage contains a part, quantity, and XYZ locations (the number of XYZ locations is equal to quantity) The lists of consumed parts and produced parts can contain 0, 1 or more items. A part can be in the consumed list as well as the produced list. When the operation executes in the Quest simulation, the consumed parts will disappear and the produced parts will appear. If the produced parts are going to be consumed in a successor operation, they will immediately go to where that successor operation is supposed to take place. If the produced parts are not going to be consumed by one of the successor operations, then they are assumed to be final parts and will stack at the location specified in the part usage object. *(Note: if produced parts are not used by an immediate successor, then they cannot be used by a remote successor: They are final parts).*

The Quest graphical user interface has been modified to allow the user to enter consumed parts and produced parts, but not to enter quantity or location for each part. This can be done in Query Manager. The user interface has also been modified to display consumed and produced part information (their names, quantity and location), The Quest GUI cannot modify consumed and produced parts.

3.4 Locations for Consumed Parts and Produced Parts

The final phase IDL specifies, for each operation, the location of where the consumed parts and produced parts are to be displayed during that operation. The location of a part is specified in the form of a transform matrix in the SAVE database. The Deneb parser extracts the XYZ coordinates from this transform matrix. These are the first three numbers in the bottom row of the 4X4 transform matrix. QUEST interprets this XYZ position as relative to the "machine's" origin.

The part locations for produced parts are only meaningful, and therefore only used, for *final* parts. (Since if it is not a final part, it will be consumed by a successor operation, and will therefore go to the part location specified in the consumed parts list of the successor operation).

When an operation consumes more than one unit of a part type, the SAVE database specifies a part location for each unit. However, Quest currently is not easily able to handle different locations for units of the same part class. Quest will assign the same part location for all units of the same part class. This location is the first location specified in the list of part locations in the SAVE database. There exists a way around this possible in future implementations.

3.5 Manufacturing Orders

For the final phase of the SAVE project, the database model was modified so that multiple manufacturing orders are associated with a process plan. The Quest parser requires that there is at least one manufacturing order in the process plan, or it will not run the simulation. Quest uses the following fields in a manufacturing order: Quantity and the Planned Start Date (contained in the SchedInfo object). The quantity must be at least 1 and the planned start date must be a valid date for the simulation to run.

The Quest parser determines the earliest among the planned start dates of all manufacturing orders, and starts the simulation from that point in time (which is internally time 0).

3.6 Work Calendars

The final phase implementation has been modified to use work calendars. Each labor can have a sequence of calendars associated with it. Each calendar corresponds to a work year, and specifies the work days of the year. The sequence of calendars then represents all the days that a labor will work, and therefore the labor, in the simulation*, will not work in a year past the list of work calendars specified for him/her*. For each day of each calendar, if it is a workday, then the worker will work the shift that is assigned to him/her in the shift field. If it is not a workday, then the worker will have a dummy shift associated with that day which involves no work. If no shift information is specified for that labor, then the labor will work constantly, and the calendar information is not used (i.e. it doesn't matter whether the calendar information is specified or not in this case). If there is shift information, but no calendar information, the worker will work all days of the year using that shift.

When there are work calendars, the Quest parser creates a schedule file for each work calendar. These files will be placed in a directory specified in one new user configurable variable, named SCHEDULES_DIRECTORY. The schedule files specifie all the days of all the work calendars, starting not from January 01st, but from earliest planned start date among all manufacturing orders, and go all the way to December 31st of the last work calendar in the sequence of work calendar for each labor.

3.7 New Configuration Variables

SCHEDULES_DIRECTORY

This is the directory where the user would like the schedules files, which correspond to the work calendars, to be placed.

OBJECT_REF_MODE

This variable can be set to either 0, 1 or 2. It is basically the answer to the question: what uniquely identifies an object, such as an operation, in the SAVE database? Is it

1= its name; 2 = id-name-description combination; or 3 = none of the above, instead, a “stringified” CORBA object reference. It is important to know what object reference mode is being used when entering data into Query Manager. We recommend to use the name as a uniquely identifying an operation. This seems to be that this is what is menat to be in QM, but it does not appear to be working properly. In other words, if I am entering OP10 data, and then OP20 data into the SAVE database using QM. Then, when editing OP20, I want to specify OP10 as a precedent for OP20. It seems that specifying the name OP10 is meant to be sufficient to refer to OP10 that I just entered. However, when displaying OP10 from the precedent list of OP20, the QM does not seem to really “remember” or “know” that I meant OP10 that I just entered (e.g. the description field of OP10 will be displayed as blank, even though I entered data into it).

3.8 CAD Model

The final phase IDL associates a list of CAD models to a part or machine. These are CAD models which represent the same geometry but in different formats. This is in order to allow various tools to use the CAD model of the format that they require. The Deneb parsers require CAD models of format “deneb”.

Appendix H

VSA 3D Wrapper User's Guide

EAI

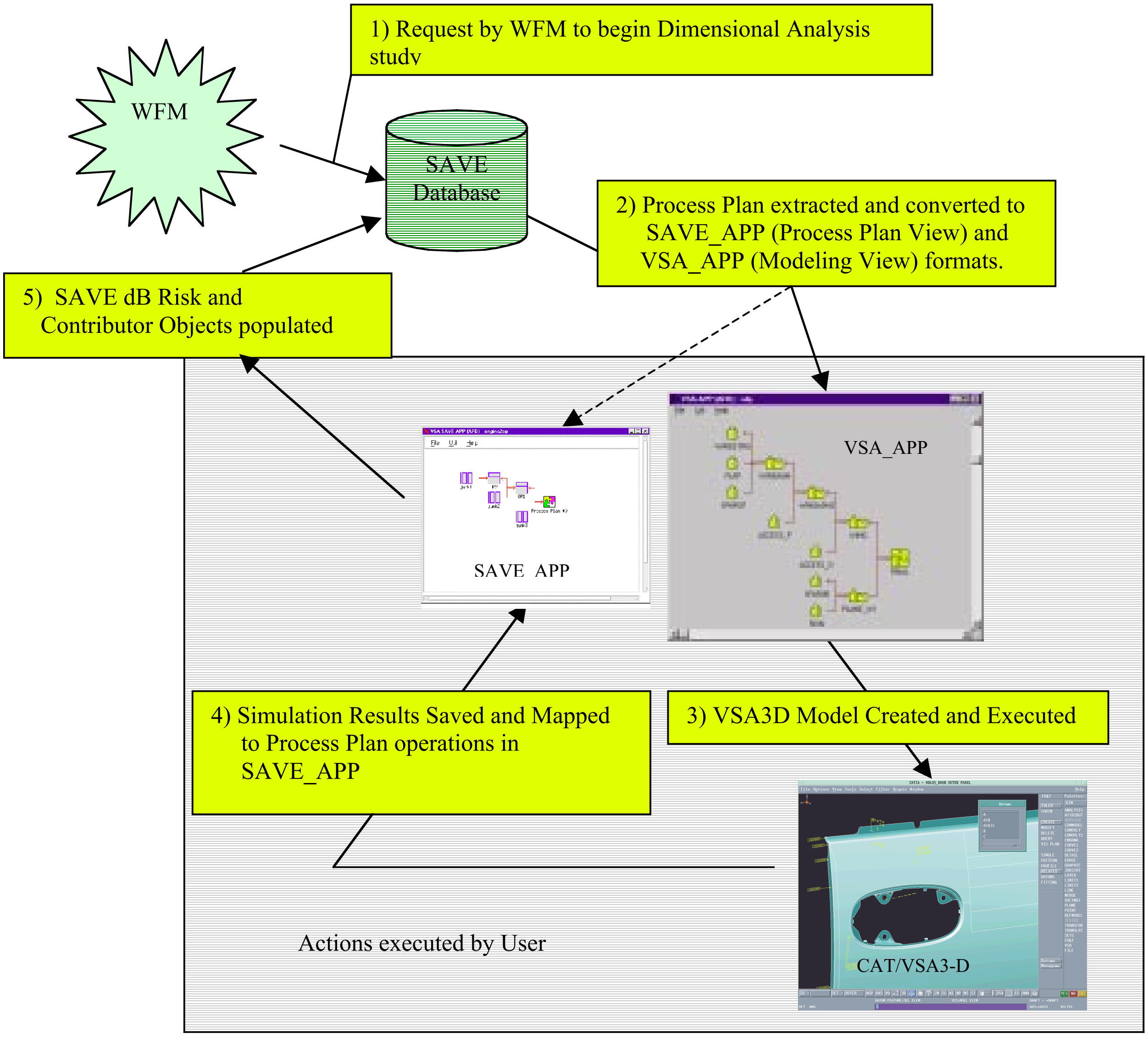
SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

1.0 Overview

The VSA clients are used to generate a graphical view of the SAVE process plan, generate a component view of the SAVE Process Plan and populate the RISK and CONTRIBUTOR objects in the SAVE Database. The population of these objects requires a Variation Simulation Model be created and executed by EAI's CAT/VSA-3D software. The component view of the process plan represents the VSA-3D AFD. The AFD is used to define modeling content and sequence.

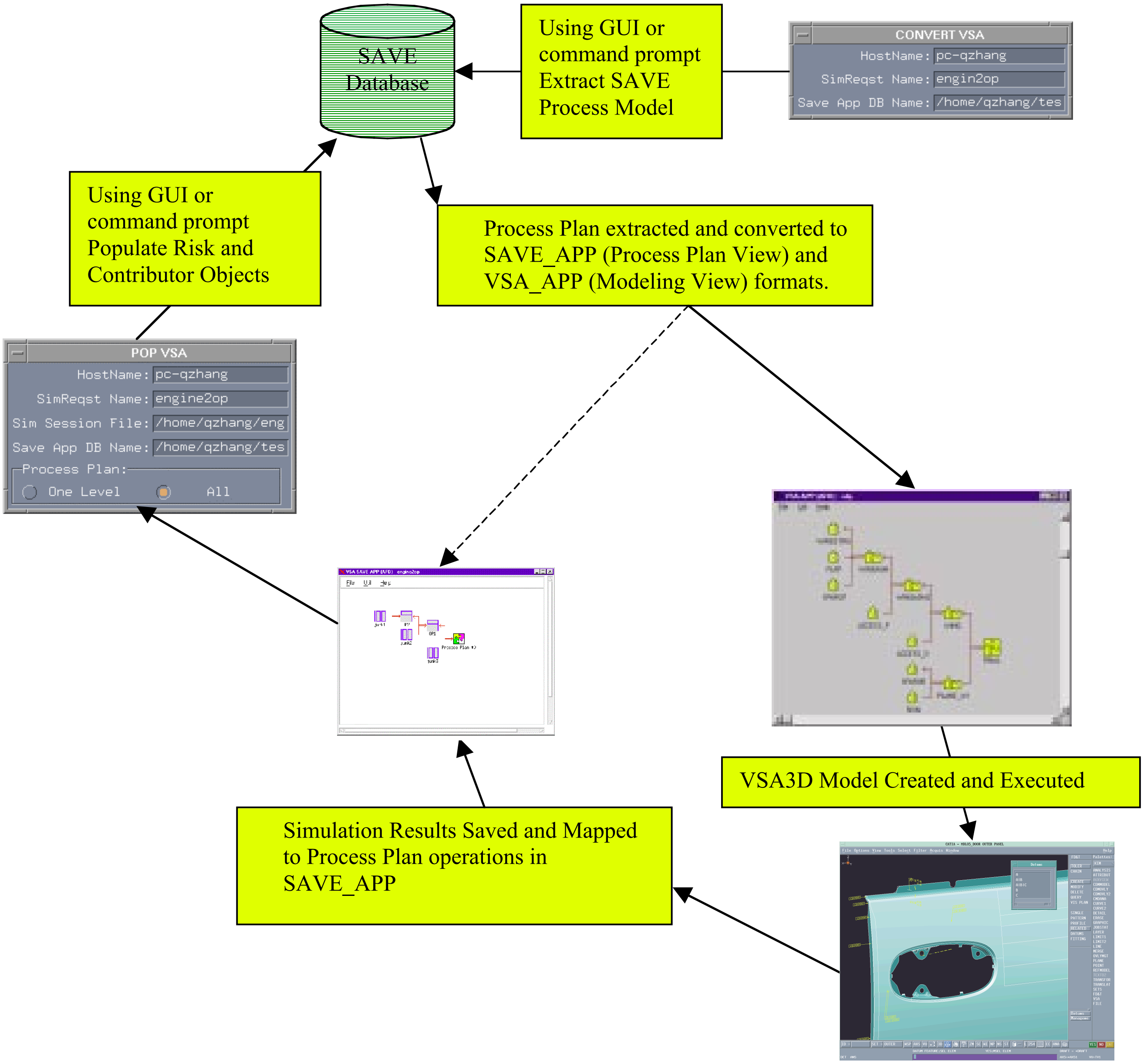
EAI has provided software clients which can be enabled by the SAVE Work Flow Manager (WFM), by using a graphical interface enabled in the CATIA environment, or by executing the clients in a command line mode. The enabling of the interaction of the VSA clients with the WFM reflects the maturation of the original intent. The Beta release of the clients, required command line operation. There was inherent benefit for continuing to support the operation of the clients outside the WFM, so a GUI was introduced to further simplify their use. This GUI can be invoked from within the CATIA environment.

The process flow can best be described using the following diagrams:


1) Request by WFM to begin Dimensional Analysis study
WFM
SAVE Database
2) Process Plan extracted and converted to SAVE_APP (Process Plan View) and VSA_APP (Modeling View) formats.
5) SAVE dB Risk and Contributor Objects populated
SAVE APP
VSA_APP
4) Simulation Results Saved and Mapped to Process Plan operations in SAVE_APP
3) VSA3D Model Created and Executed
Actions executed by User
CAT/VSA3-D

Flow Using Work Flow Manager

Alternatively, the extraction of the process plan and population of the SAVE database can be completed by executing the VSA clients (convert and pop) either from a CATIA GUI or by executing the clients on the command line.



SAVE Database

Using GUI or command prompt Extract SAVE Process Model

CONVERT VSA

HostName: pc-qzhang

SimReqst Name: engin2op

Save App DB Name: /home/qzhang/tes

Using GUI or command prompt Populate Risk and Contributor Objects

Process Plan extracted and converted to SAVE_APP (Process Plan View) and VSA_APP (Modeling View) formats.

POP VSA

HostName: pc-qzhang

SimReqst Name: engine2op

Sim Session File: /home/qzhang/eng

Save App DB Name: /home/qzhang/tes

Process Plan:

○ One Level ● All

VSA3D Model Created and Executed

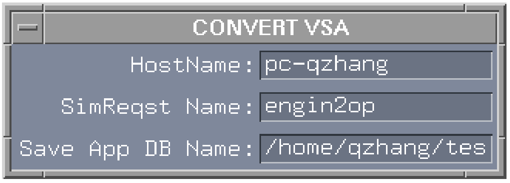
Simulation Results Saved and Mapped to Process Plan operations in SAVE_APP

Flow Without Using Work Flow Manager

2.0 VSA SAVE Client Overview

This section contains a brief overview of the input and output for each of the three VSA SAVE clients contained in the release 1.0 shipment.

2.1 Convert_VSA



HostName: pc-qzhang

SimReqst Name: engin2op

Save App DB Name: /home/qzhang/tes

Command Prompt Usage: convert_vsa <hostname> < SimReq name > <Save_APP dB Name>

- ***extracts SAVE PROCESS PLAN to a SAVE_APP dB file***

- ***creates VSA_APP file***

- ***the SAVE_APP db_file is read by save_app to graphically view the PROCESS PLAN and associate VSA measurements with SAVE OPERATIONS***

- the VSA_APP file is read by the VSA_APP to graphically view the components and operations in a Variation Simulation Analysis. It is used in the modeling process to define content and sequence.

2.1.1 Input

1. Host name of machine on which SAVE data base server resides

2. Name of simulation request object

3. Name of the SAVE_APP dB file to create. The name provided is the name given to the VSA_APP file as well.

2.1.2 Output

1. SAVE_APP dB File

2. VSA-APP File.

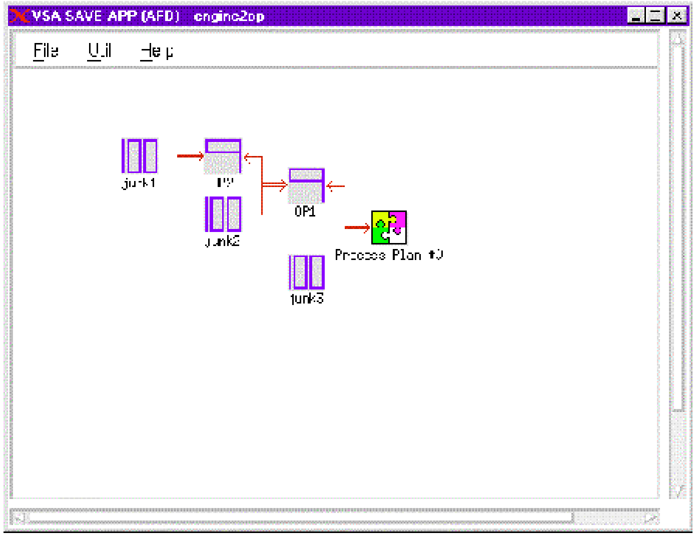
2.1.3 Example

➢ convert_vsa professor LearningAide vsatest.hst

(*we recommend the use of ".hst" file extension since SAVE_APP looks for this as default*)

Distribution Statement A: Approved for public release. Distribution is unlimited.

2.2 SAVE_APP



- ***graphically view the PROCESS PLAN and associate VSA measurements with SAVE OPERATIONS***
- save an updated SAVE_APP dB file which is used by the pop_vsa client to define which VSA measurements are used to calculate risk and contributor information for a SAVE OPERATION

2.2.1 Input

1. SAVE_APP dB File created from convert_vsa program
2. Link to measurement files from the VSA-3D study

NOTE: *The name of the measurement file must be the name of the measurement. This is the default naming convention used by CAT VSA-3D.*

2.2.2 Output

1 . updated SAVE_APP dB File with link between VSA measurements and SAVE OPERATIONS

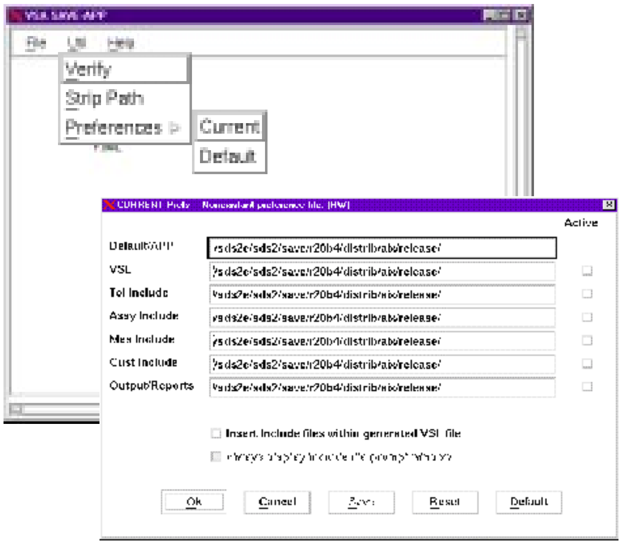
Hint: We recommend saving the SAVE_APP dB File in the directory containing your VSA-3D model files to reduce traversal of directories when using SAVE_APP.

2.2.3 Using Save_APP

1. It is recommended that the working directory be set using ***UTIL⇒Preferences⇒Current.***

   Set the ***default directory*** to the location of your VSA-3D model files (and SAVE_APP dB File).

   When running the SAVE_APP, you will be selecting measurement files (exactly as done in the VSA-APP), setting the default directory will save time traversing directories to select files.



		Active
DefaultAPP	/sds2e/sds2/save/r20b4/distrib/aix/release/	
VSL	/sds2e/sds2/save/r20b4/distrib/aix/release/	□
Tol Include	/sds2e/sds2/save/r20b4/distrib/aix/release/	□
Assy Include	/sds2e/sds2/save/r20b4/distrib/aix/release/	□
Mes Include	/sds2e/sds2/save/r20b4/distrib/aix/release/	□
Cust Include	/sds2e/sds2/save/r20b4/distrib/aix/release/	□
Output/Reports	/sds2e/sds2/save/r20b4/distrib/aix/release/	□

□ Insert Include files within generated VSL file

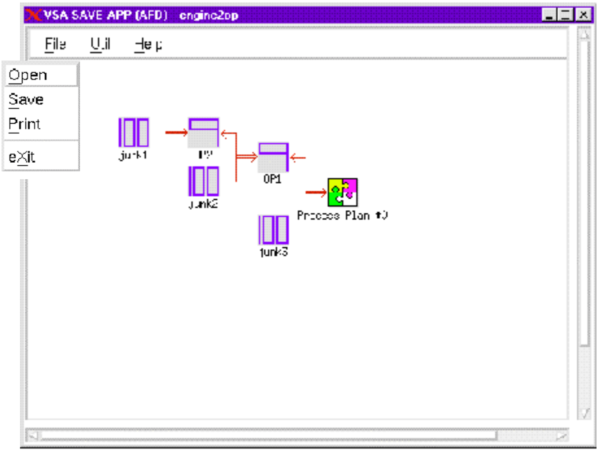
□ Always display include file prompt window

2. The next step is to select the SAVE_APP dB File created from your SAVE PROCESS PLAN. In our example, we use a small test program.

   Hint: If you know that the simulation study is focused on only a portion of the overall SAVE PROCESS PLAN, we strongly recommend creating a simulation request object for the PROCESS PLAN which represents the local area of interest.

   This will greatly reduce the number of operations you need to view and traverse to properly assign VSA measurements to your SAVE OPERATIONS of interest!

   Use the ***File⇒Open*** menu to select the SAVE_APP dB File. Notice that the default filter extension is "*.hst" to filter out all other files in your directory. Once you select the hostfile, your graphical display will update to look something like this.

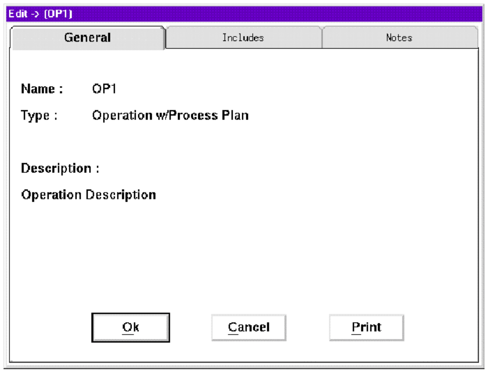


3. ICON graphics show a number "9" for a Process Plan and number "100" for individual Operations.

4. Double clicking on any icon will access the properties for each operation.

The ***General*** panel contains:

- the operation name and description



Name : OP1

Type : Operation w/Process Plan

Description :

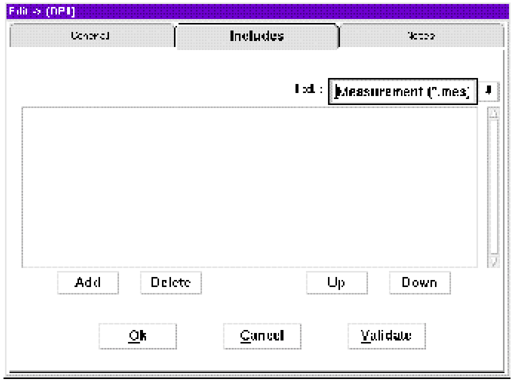
Operation Description

The ***Includes*** panel allows you to:

- associate VSA-3D measurement files to SAVE operations

  (The name of the measurement must be the same as the measurement file name)

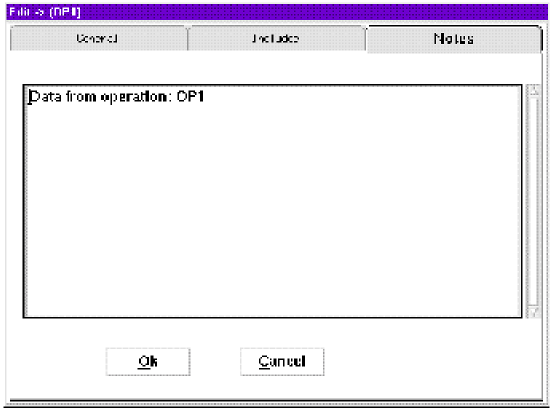
- remove VSA measurements from your SAVE OPERATION
- validate whether the selected files are accessable from save_app
- reorder the list of measurement files



List : Measurement (*.mes)

The ***Notes*** panel displays the following operation attributes:

- Operation Type
- Consumed Part(s)
- CAD reference to Consumed Part(s)
- Produced Part
- CAD reference to Produced Part



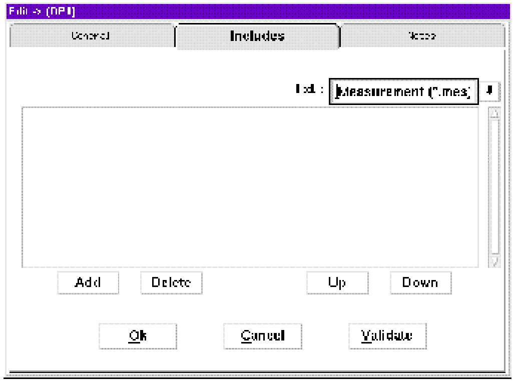
Data from operation: OP1

To associate a VSA mesurement to an Operation:

- Select the Add Button

  When the add button is selected, the file selection menu shown on the right is presented.

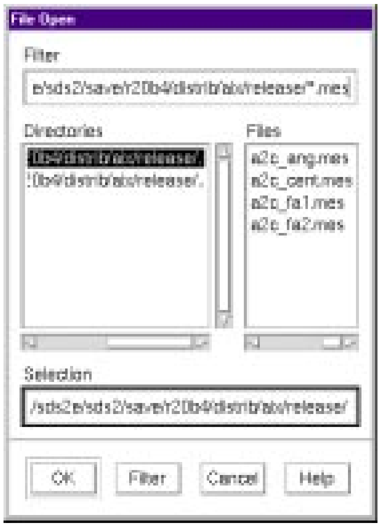
  The default directory path is defined in the preferences menu as described earlier.



Ext : Measurement (*.mes;

  The file filter is defaulted to files with the extension “*.mes”. This is the extension for measurements created using the CAT VSA-3D application.

- Select a VSA measurement by double clicking the file in the selection box in the Files listing. Change directories by double clicking on directories in the Directories listing.



Filter
e/sds2/save/r20b4/distrib/abi/release/*.mes

Directories
/0b4/distrib/abi/release/.
/0b4/distrib/abi/release/..

Files
a2c_ang.mes
a2c_cent.mes
a2c_fa1.mes
a2c_fa2.mes

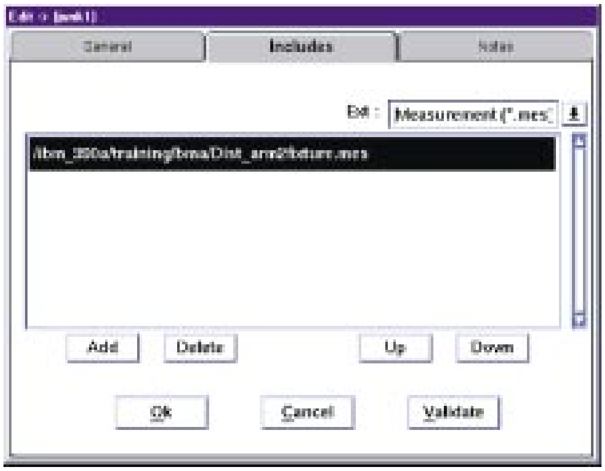
Selection
/sds2e/sds2/save/r20b4/distrib/abi/release/

- The selection will be display on the Include panel

5. To remove the association of a VSA measurement and an Operation

- Highlight the measurment on the Include panel

- Select the delete button

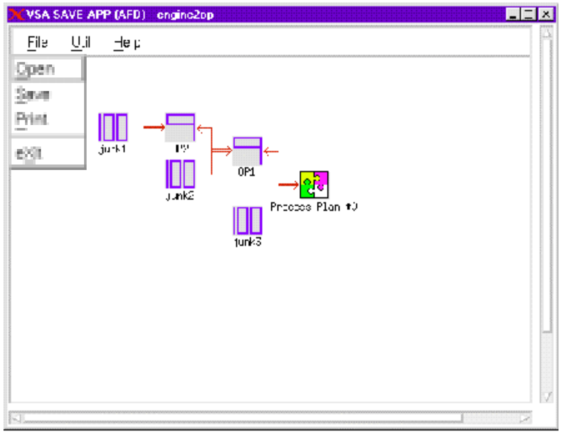


Ext : Measurement (*.mes;

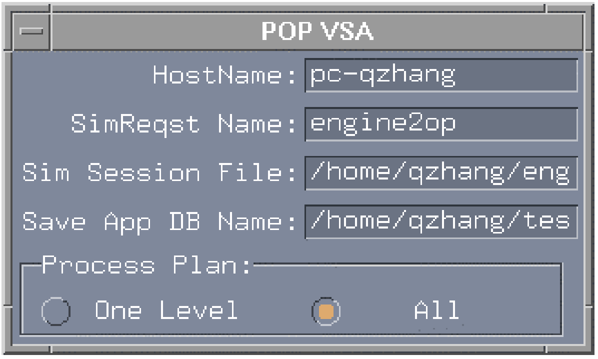
/ibm_300a/training/bmw/Dist_arm2fixture.mes

6. Once all necessary relationships between VSA measurements and SAVE OPERATIONS have been established, save the updates to the SAVE_APP dB File. From the main save_app window using ***File⇒Save*** from the main window.

7. Selection of ***File⇒Study Complete*** should now be selected to signal the WFM that the analysis activities have been completed.



2.3 POP_VSA



POP VSA

HostName: pc-qzhang

SimReqst Name: engine2op

Sim Session File: /home/qzhang/eng

Save App DB Name: /home/qzhang/tes

Process Plan:

○ One Level ● All

Command Prompt Usage:

pop_vsa <hostname> <SimReq name> <VSA Sim Session file > <hostfile> [-all]

- ***extracts risk and contributor information from a VSA-3D session file and saves it to the Operation Risk and Contributor objects***

- ***the SAVE_APP dB File contains links from VSA-3D measurement(s) to the SAVE OPERATION***

- ***risk and contributor information is calculated from ALL measurements associated with the SAVE OPERATION***

2.3.1 Input

1. Host name of machine on which SAVE data base server resides

2. Name of simulation request object to in initiate the VSA SAVE client request

3. Name of the VSA-3D simulation model file from which to extract RISK and CONTRIBUTOR information

4. SAVE_APP dB File name

Note: Session files are created from the VSA-SIM analysis program. Session files have the file extension ".sim".

5. Optional [-all] option to force traversal of all levels of process plan. Without the [-all] option, only a single level of the process plan is traversed.

2.3.2 Output

• Risk Object(s)	(associated to Operation object(s))
• Contributor Object(s)	(associated to Risk object(s))
• Simulation Model Object containing full path and name of the VSA-3D session file	(associated to Process Plan object)

2.3.3 Example

➢ pop_vsa professor LearningAide vsatest.sim vsatest.hst

(populate the process plan referenced by simulation request object LearningAide in SAVE database on server professor with the risk and contributor information from vsatest.sim using the vsatest.hst file to define mapping between VSA measurements and SAVE database operations)

Pop_vsa calculates risk and contributor information for a SAVE operation with measurements associated with that operation using ***save_app***.

Pop_vsa matches operations in the first level of the SAVE process plan (contained in the SimReq object).

The ***pop_vsa*** client will not traverse into the second level of a process plan to find a matching operation name. It will only match operation names at the first level of the Process Plan contained in the SimReq object. ***Pop_vsa*** lists each of the operations and which measurements (if any) were used to populate risk and contributor information.

VSA SAVE Client supports recursive searching for operation names if the [-all] flag is provided. (The square brackets are not typed, but used to denote an optional parameter) This will support finding and matching operations in the given process plan and in all process plans referred to by the original.

Appendix I

CATIA™ Costlink User's Guide
Cognition Corporation

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

1.0 Overview

Costlink-CT is a tool developed by Cognition Corporation, which extracts computer-aided design (CAD) model feature data from CATIA™ and passes it into the Cost Advantage™ cost estimating tool. Methods were developed for extracting both assembly and component information. Inputs from three (3) companies were utilized to create the specification for the latest version of Costlink. This industry-generated specification allows the SAVE demonstration system to be a basis for future commercial products versus a single application demonstration tool. The Costlink concept is also applicable to other mechanical and electronic CAD tools.

This tool provides significant benefit to the user community and cost estimating process because it allows the design engineer (designer) to pass design information to the cost estimator directly, precluding manual extraction from a drawing. A drawback of this tool, however, is that it only picks up design features that have been pre-defined as cost drivers. Thus, the designer and cost estimator must still communicate with each other about unusual aspects of the new design.

2.0 Concept of Operation

A designer begins the process using the company's "Best Practices" to design a part using the CAD tool CATIA™. As shown in Figure I-1, standard conventions for design features need to be utilized to gain optimum benefits from this integrated system.

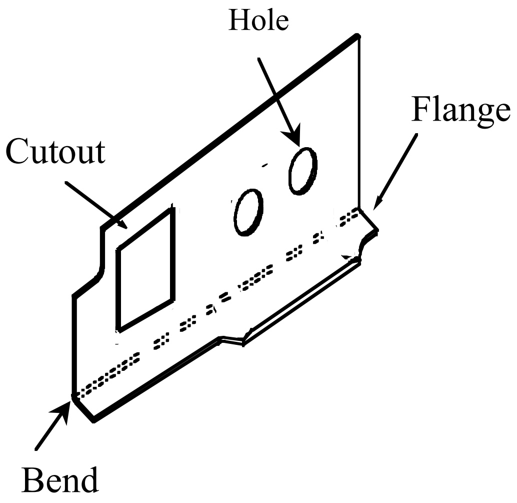


Figure I-1: Standard Feature Definition Example

The designer meets with the rest of the SAVE team to discuss the component or assembly being developed to maximize the Design for Manufacturing (DFM) benefits. When the component or assembly is sufficiently complete in the CAD tool, the designer invokes the Costlink function from within CATIA™. This process will extract the design data directly into the Cognition Knowledge Center (KC) object base. This design data is then passed to the Cost Advantage™ cost estimating session via a set of scripts in the KC. Scenarios for using the integrated SAVE system and this design data to generate a cost estimate are described in more detail in Appendix J.

3.0 Capabilities of the CATIA™ Costlink

The CATIA™ Costlink to Cost Advantage™ was designed to utilize both the basic CATIA™ capability and a company's “Best Practices”. While the Costlink itself pulls design information from CATIA™, the ability to match that data to the cost estimating relationships in the cost model is dependent on using these Best Practices. This is an area where an individual company can customize its tools to support its methods. The system supports both individual components and assemblies, such as shown in Figure I-2. The data extracted from CATIA™ is stored in the Knowledge Center (KC), where it can be utilized by both Cost Advantage™ and other systems. To support a more flexible environment, this data is not interpreted in the Costlink module, rather through Basic scripts in the Knowledge Center. This is an enhancement from the interim SAVE demonstration version.

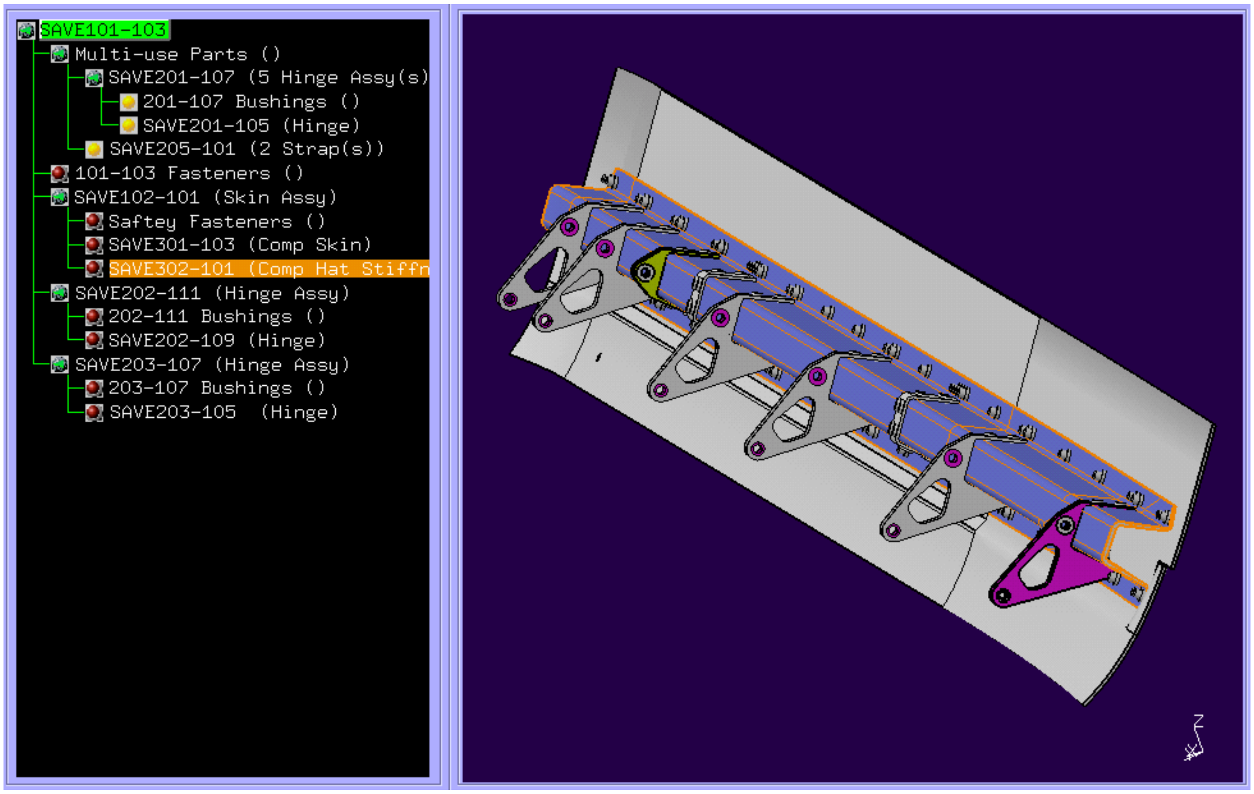


Figure I-2: SAVE Door Assembly CAD Example

The Assembly Costlink is currently developed for CATIA™ version 4.1.9 on the IBM AIX platform. A new revision of Costlink for a future version of CATIA™ is currently being developed by Cognition for commercialization, based on lessons learned from the SAVE program. A detailed description of the current Costlink capabilities and future enhancements may be found in the Cognition Corporation product specification.

In the SAVE demonstration version, look-up tables for Parts and Fasteners have been created to provide design data for the cost estimating model which is not readily available from the CAD tool, or which is typically not placed in the CAD tool at the time that the estimates are made. The latest available design data will always be used for a cost session; e.g., if "Material Type" were available in both a look-up table and the CAD model output, the CAD model data would be used. These tables (and associated KC scripts) can be tailored to meet the needs of the company's product.

4.0 SAVE Demonstration Assembly Costlink User Instructions

The following directions are for the demonstration version of the Costlink tool. The associated data flow is shown in Figure I-3. The commercial version being developed by Cognition Corporation will be more automated.

- Design the parts and assemblies in CATIA™ utilizing the company's “Best Practices”.
- Invoke Costlink from CATIA™ and save the design data in the Knowledge Center.
- Within Cost Advantage™, either retrieve the Cost Note for the trade study being performed, or use a SAVE generated note. Make modifications, and save the updated note to the Knowledge Center.
- In the Knowledge Center, select the Cost Note that was previously created in Cost Advantage™. Select Costlink plus any applicable database table inputs needed for the cost model.
- In Cost Advantage™, restore this updated Cost Note and continue cost estimating studies. Save the results.
- Save the pertinent data from the Cost Note to the SAVE database.
- Modify the engineering model to reflect the cost estimating and other SAVE application specific results.
- Repeat this process as required.

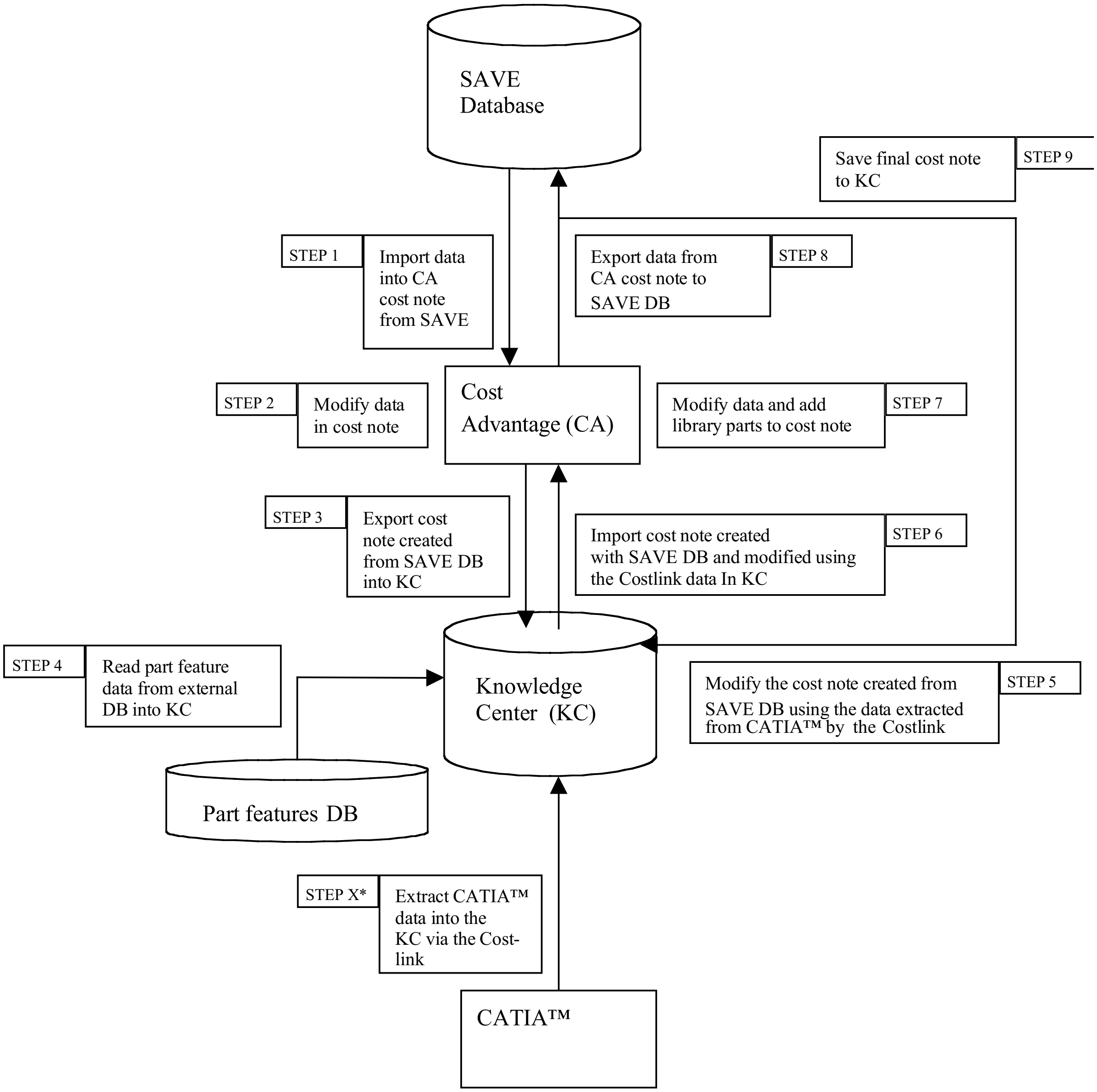

SAVE Database
Save final cost note to KC
STEP 9
STEP 1
Import data into CA cost note from SAVE
Export data from CA cost note to SAVE DB
STEP 8
STEP 2
Modify data in cost note
Cost Advantage (CA)
Modify data and add library parts to cost note
STEP 7
STEP 3
Export cost note created from SAVE DB into KC
Import cost note created with SAVE DB and modified using the Costlink data In KC
STEP 6
STEP 4
Read part feature data from external DB into KC
Knowledge Center (KC)
Modify the cost note created from SAVE DB using the data extracted from CATIA™ by the Costlink
STEP 5
Part features DB
STEP X*
Extract CATIA™ data into the KC via the Cost-link
CATIA™

Figure I-3: Cost Advantage™ / SAVE / Knowledge Center Flow and Data Interactions

5.0 Costlink Architecture and Data Flow

The system architecture associated with the Costlink to Cost Advantage™ interactions is shown in Figure I-4. Additional information about the Costlink is available from Cognition Corporation, the system developer.

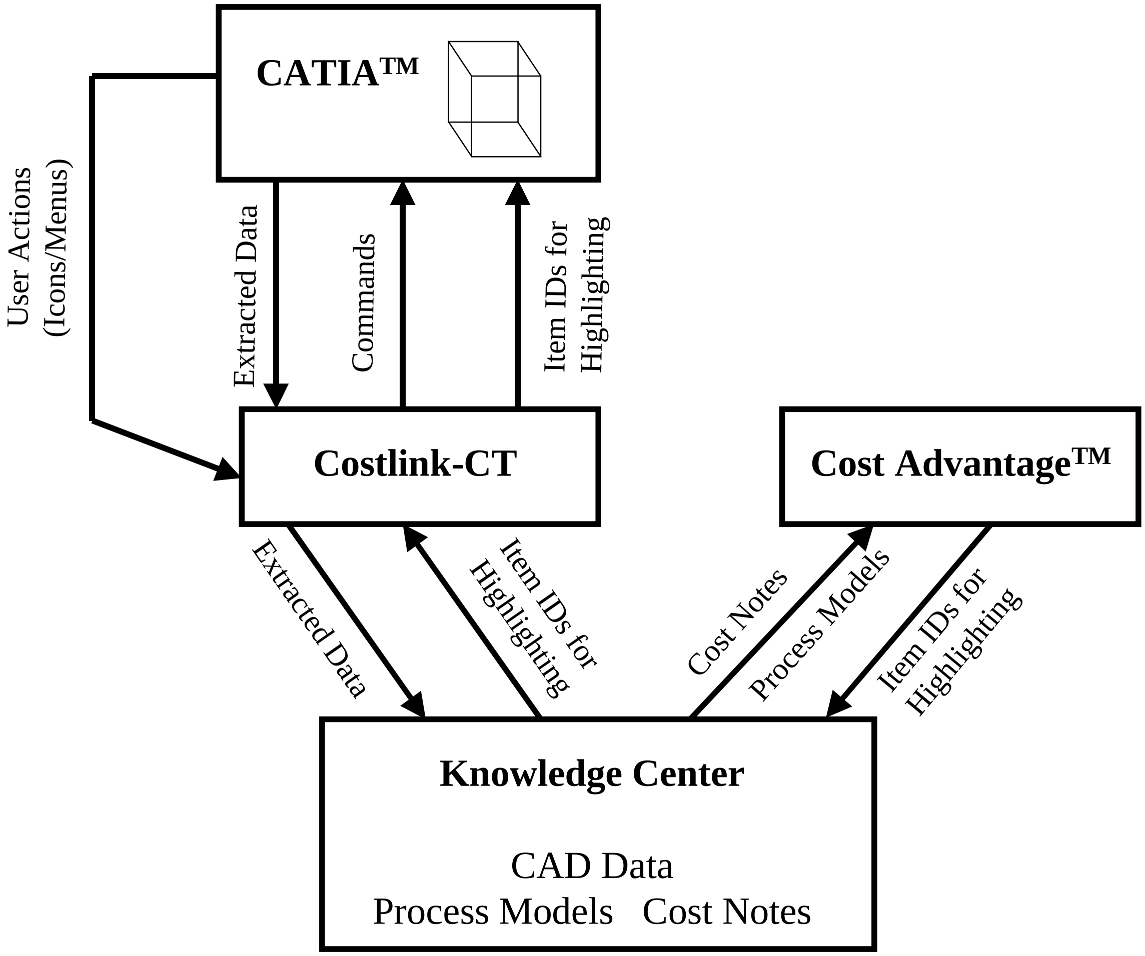

CATIA™
User Actions (Icons/Menus)
Extracted Data
Commands
Item IDs for Highlighting
Costlink-CT
Cost Advantage™
Extracted Data
Item IDs for Highlighting
Cost Notes
Process Models
Item IDs for Highlighting
Knowledge Center
CAD Data
Process Models Cost Notes

Figure I-4: Costlink Architecture

Appendix J

Cost Model User's Guide

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

1.0 Overview – General Approach

The cost estimating tool within SAVE extends the capability of traditional cost estimating tools to integrate outputs from manufacturing simulation tools and CAD tools. This provides a more robust cost estimate that is based on both design features and the manufacturing processes utilized to produce the component. The SAVE cost estimating models will take design, business, and cost inputs and utilize the programmed expertise to provide cost and producibility guidance to the design team. Information from multiple sources is integrated to provide the cost estimate as described in Figure J-1.

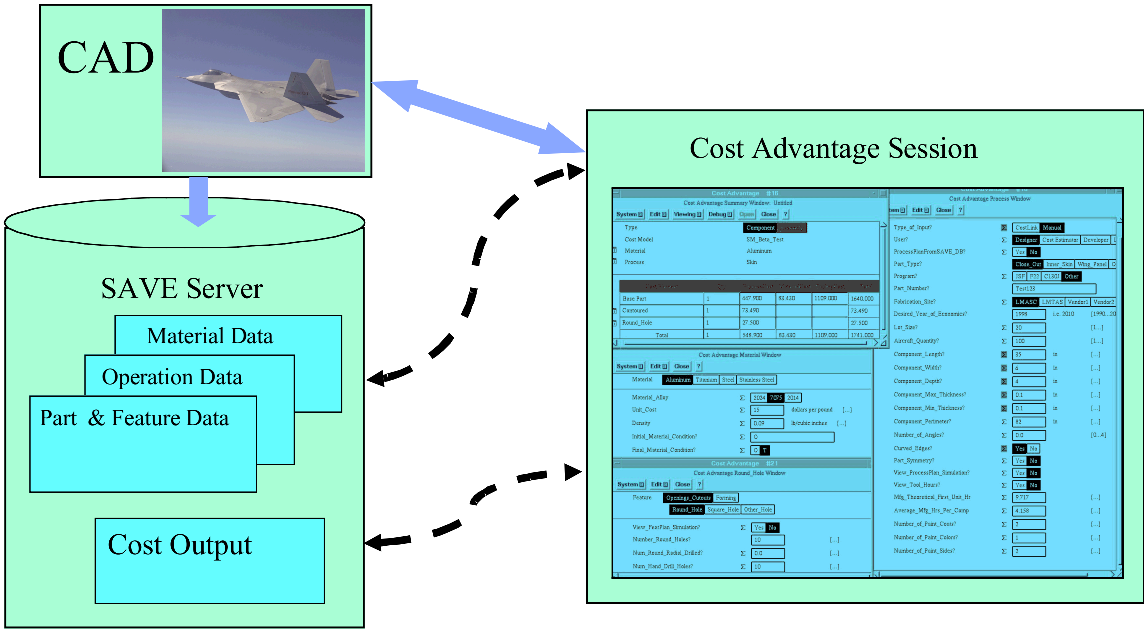


Figure J-1: Data Shared Among Cost Module, CAD, and SAVE Server

A key aspect of this cost estimating method is its capability to relate product features to manufacturing processes. Each company can customize its cost model to add features that are cost drivers in its manufacturing environment. Since this cost tool is designed as a shell, a company's specific cost algorithms, help screens, and rules can also be added. Figure J-2 describes top-level inputs and outputs of the cost estimating system. Both automated SAVE system inputs and cost estimator user inputs are utilized. The output is an estimate of the cost of producing the part or assembly.

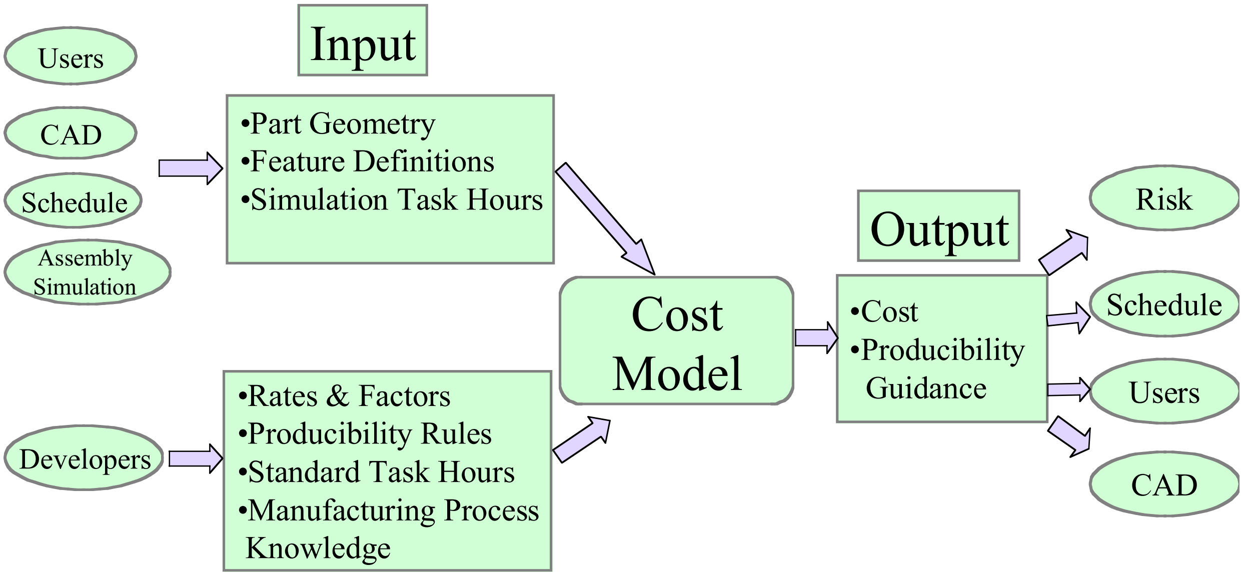


Figure J-2: SAVE Cost Model – Input and Output

For the SAVE program, Cost Advantage™ (CA) models were built to accommodate requirements for assembly, sheet metal, numerically controlled (NC) machining, and hand lay-up composite parts. Representative operations and cost estimating relationships (CERs) are included as a starting point for future development.

2.0 Concept of Operation

2.1 System Users

The SAVE Cost Advantage™ tool will be accessed by multiple members of the team. Each person will have a different viewpoint of what data he wants to see. For example, the cost estimator, who is the primary user of the system, will develop cost estimates for the design trade study using both the expert knowledge embedded in the system and his personal expertise. He may also be modifying learning curve factors and labor rates and factors.

The design engineer will utilize the system to obtain a quick look at the cost impact of his design when his design is within the bounds of the cost model. This occurs for most derivatives, and conventional parts. Figure J-3 shows the diversity of users interacting with these cost estimating models, both through supplying information for developing cost estimating relationships to the developer and as end users.

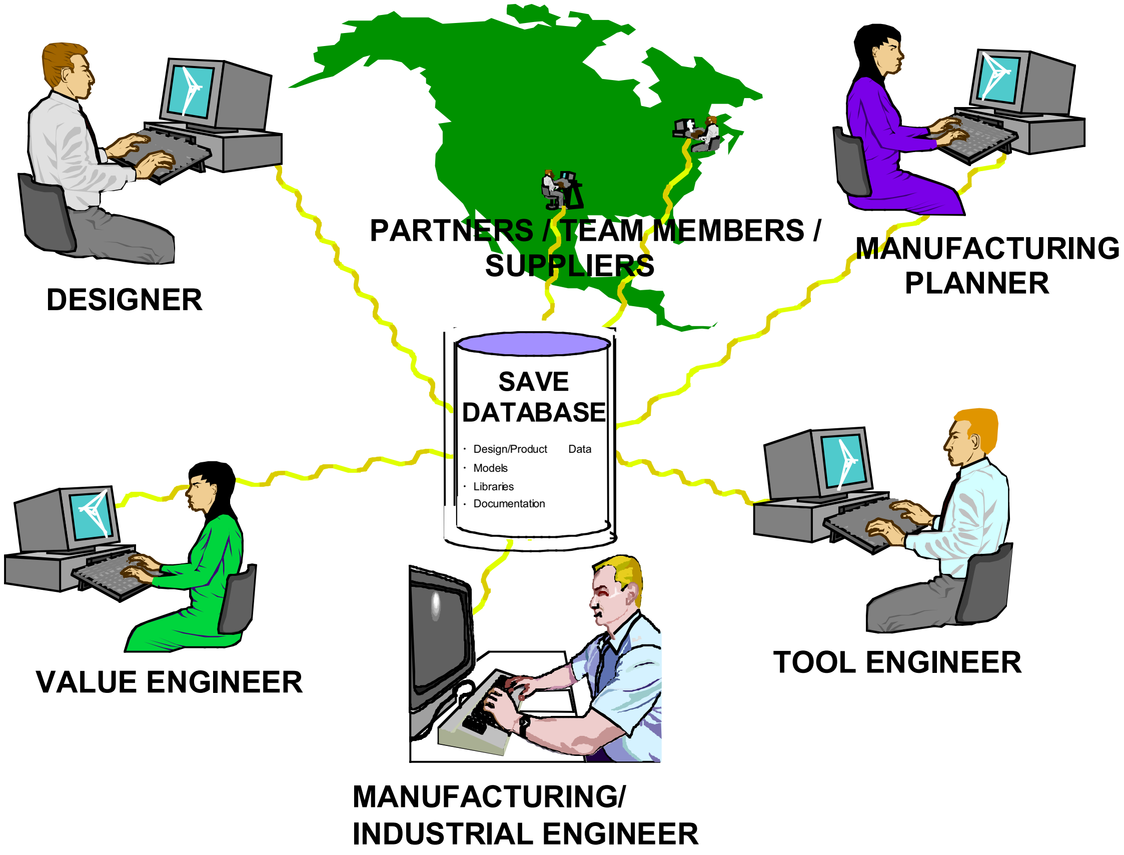


Figure J-3: Personnel Resources Required for the SAVE Cost Model

2.2 Cost Estimating Model Usage Scenarios

2.2.1 SAVE Team Accessing Cost Model from the SAVE Environment

In this scenario, the cost estimator would utilize standard SAVE practices to start the Cost Advantage™ application and extract the SAVE data base information into the cost model. Design data from CATIA™ is acquired from Costlink via the Knowledge Center and Cost Advantage™ Cost Notes. The ordering and interactions for performing these steps are included in the CATIA™ Costlink User's Guide, Appendix I.

In the Assembly cost model, cost estimates can be generated either from simulation hours obtained from the SAVE simulation runs, or from calculated hours pre-defined in the cost model. This allows the rich knowledge from the simulations to be incorporated into the final cost estimating results. Process plan steps are also imported from SAVE to reflect the current understanding of the manufacturing process.

2.2.2 Design Engineer Accessing Cost Model from the CAD Tool

Design engineers performing initial design trade studies are key users of the system. Other Integrated Product Team (IPT) members, such as cost estimators and manufacturing engineers, will be supporting the designers in their efforts.

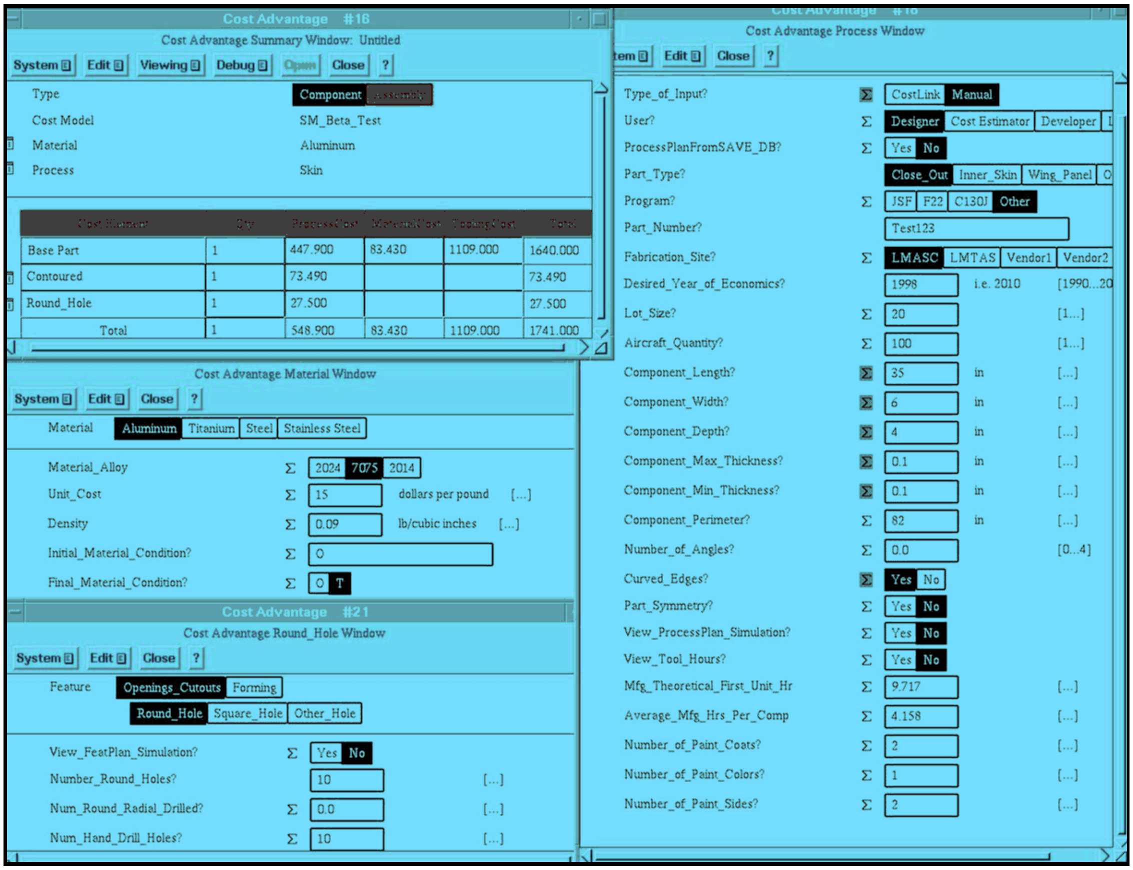
This scenario is where the design engineer is working in CATIA™ and would like a quick idea of how much the component or assembly costs. At this point, the designer needs to be aware that the result is a preliminary estimate, because all features in the component have not yet been defined. The designer selects "costing" as an option in the CAD session. Cost Advantage™ is then initiated from within the CAD session. Cost Advantage™ accepts part geometry from the CAD tool, CATIA™, through the Costlink module and delivers immediate cost and producibility feedback to the designer. This information is also stored in the SAVE database using the export function in the Cost Advantage™ wrapper. High cost drivers are revealed as they are introduced and alternatives are available to offer cost reduction or increased product reliability. The output of the cost-estimating tool provides the primary cost drivers in the assessment of design alternatives.

Best design practices such as feature libraries and standards must be utilized for optimum benefit of this tool. Only those design features that are pre-defined in the cost model provide cost outputs to the user. The cost model developers will have accounted for key design cost drivers as defined by the company's design "Best Practices".

2.2.3 Cost Estimator or Value Engineer using the Cost Model in a Stand Alone Mode

This is the traditional method for using Cost Advantage™. The cost estimator will either manually input data into Cost Advantage™, or retrieve a previously completed Cost Advantage™ trade study as a starting point. This is a good environment for conducting "what-if" trades on non-SAVE related characteristics, such as material types or learning curve slopes. The user can conduct these trades with support from other IPT members, and use this knowledge in later SAVE sessions.

This method is useful if the cost estimator is not in a SAVE environment, and does not have access to CAD models. Use the "TypeOfInput" variable set to "Manual" in the assembly and sheet metal cost models to view additional inputs required for costing the parts. Figure J-4 shows a typical user's screen that would be seen in all of the user scenarios.



Cost Advantage Summary Window: Untitled

Type [✓ Component] Assembly

Cost Model SM_Beta_Test

Material Aluminum

Process Skin

Cost Element	Qty	ProcessCost	MaterialCost	ToolingCost	Total
Base Part	1	447.900	83.430	1109.000	1640.000
Contoured	1	73.490			73.490
Round_Hole	1	27.500			27.500
Total	1	548.900	83.430	1109.000	1741.000

Cost Advantage Material Window

Material [✓ Aluminum] Titanium Steel Stainless Steel

Material_Alloy Σ 2024 [✓ 7075] 2014

Unit_Cost Σ 15 dollars per pound [...]

Density Σ 0.09 lb/cubic inches [...]

Initial_Material_Condition? Σ O

Final_Material_Condition? Σ O [✓ T]

Cost Advantage Round_Hole Window

Feature [✓ Openings_Cutouts] Forming

[✓ Round_Hole] Square_Hole Other_Hole

View_FeatPlan_Simulation? Σ Yes [✓ No]

Number_Round_Holes? 10 [...]

Num_Round_Radial_Drilled? Σ 0.0 [...]

Num_Hand_Drill_Holes? Σ 10 [...]

Cost Advantage Process Window

Type_of_Input? Σ CostLink [✓ Manual]

User? Σ [✓ Designer] Cost Estimator Developer

ProcessPlanFromSAVE_DB? Σ Yes [✓ No]

Part_Type? [✓ Close_Out] Inner_Skin Wing_Panel

Program? Σ JSF F22 C130J [✓ Other]

Part_Number? Test123

Fabrication_Site? Σ [✓ LMASC] LMTAS Vendor1 Vendor2

Desired_Year_of_Economics? 1998 i.e. 2010 [1990...20

Lot_Size? Σ 20 [1...]

Aircraft_Quantity? Σ 100 [1...]

Component_Length? Σ 35 in [...]

Component_Width? Σ 6 in [...]

Component_Depth? Σ 4 in [...]

Component_Max_Thickness? Σ 0.1 in [...]

Component_Min_Thickness? Σ 0.1 in [...]

Component_Perimeter? Σ 82 in [...]

Number_of_Angles? Σ 0.0 [0...4]

Curved_Edges? Σ [✓ Yes] No

Part_Symmetry? Σ Yes [✓ No]

View_ProcessPlan_Simulation? Σ Yes [✓ No]

View_Tool_Hours? Σ Yes [✓ No]

Mfg_Theoretical_First_Unit_Hr Σ 9.717 [...]

Average_Mfg_Hrs_Per_Comp Σ 4.158 [...]

Number_of_Paint_Coats? Σ 2 [...]

Number_of_Paint_Colors? Σ 1 [...]

Number_of_Paint_Sides? Σ 2 [...]

Figure J-4: Cost Advantage™ End User Screen Example

2.2.4 Cost Estimator or Value Engineer Updating Cost Models

Cost models are an evolving part of a business. As the environment and manufacturing processes change, the cost models need to be updated to reflect this. Labor rates and factors will also need to be updated. The Cost Estimating or Value Engineering departments are typically responsible for these model updates. They will obtain information from many other organizations and sources including:

- Industrial / Manufacturing Engineering
  - Standards
  - Manufacturing process changes
- Tool Engineering
  - Modified and new tools and fixtures
  - Tooling methods
  - Cost estimating relationships for tooling

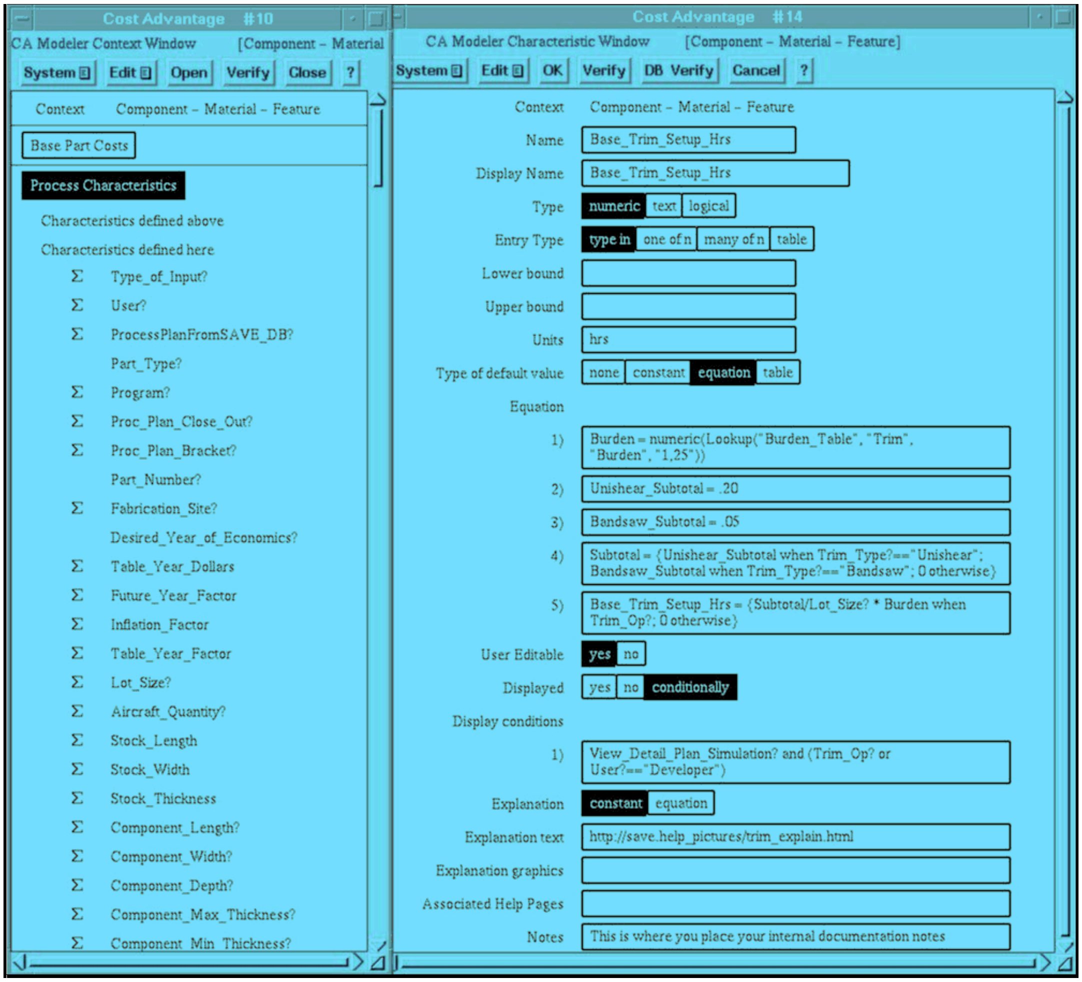
- Design Engineering
    - New key design characteristics
    - Design best practices for Costlink
- Finance
    - Labor rates
    - Inflation factors
- Manufacturing Planner
    - Current part planning and rules
- SAVE Database Administrator
    - Current IDL and data definitions

Additional information on how to update a cost estimating model is included in the Cost Model Development Guide, Chapter 6 of the Software End Item Description Document, as well as in the Cost Advantage™ User's Guide. The SAVE cost models are designed so that a company can add in its own proprietary relationships and data.

Typical information that would be modified by the developer includes:

- Proprietary cost estimating relationships (CERs)
- Additional or modified manufacturing processes
- Labor rates and factors
- Inflation factors
- Proprietary default values for variables
- Additional design features and characteristics

The Cost Advantage™ cost estimating tool is designed as an expert system shell. This allows the users to modify the existing SAVE models to reflect their own business practices and environments. The following graphic is a view of the Cost Advantage™ developer's environment. The syntax is straightforward, as shown in Figure J-5, so modifications to an existing model are very easy for a computer literate, experienced cost estimator to make.



Context Component – Material – Feature

Base Part Costs

Process Characteristics

Characteristics defined above

Characteristics defined here

Σ Type_of_Input?
Σ User?
Σ ProcessPlanFromSAVE_DB?
Part_Type?
Σ Program?
Σ Proc_Plan_Close_Out?
Σ Proc_Plan_Bracket?
Part_Number?
Σ Fabrication_Site?
Desired_Year_of_Economics?
Σ Table_Year_Dollars
Σ Future_Year_Factor
Σ Inflation_Factor
Σ Table_Year_Factor
Σ Lot_Size?
Σ Aircraft_Quantity?
Σ Stock_Length
Σ Stock_Width
Σ Stock_Thickness
Σ Component_Length?
Σ Component_Width?
Σ Component_Depth?
Σ Component_Max_Thickness?
Σ Component_Min_Thickness?

Cost Advantage #14
CA Modeler Characteristic Window [Component – Material – Feature]
System
Edit
OK
Verify
DB Verify
Cancel
?

Context: Component – Material – Feature
Name: Base_Trim_Setup_Hrs
Display Name: Base_Trim_Setup_Hrs
Type: [✓ numeric] text logical
Entry Type: [✓ type in] one of n many of n table
Lower bound:
Upper bound:
Units: hrs
Type of default value: none constant [✓ equation] table
Equation
1) Burden = numeric(Lookup("Burden_Table", "Trim", "Burden", "1.25"))
2) Unishear_Subtotal = .20
3) Bandsaw_Subtotal = .05
4) Subtotal = {Unishear_Subtotal when Trim_Type?=="Unishear"; Bandsaw_Subtotal when Trim_Type?=="Bandsaw"; 0 otherwise}
5) Base_Trim_Setup_Hrs = {Subtotal/Lot_Size? * Burden when Trim_Op?; 0 otherwise}
User Editable: [✓ yes] no
Displayed: yes no [✓ conditionally]
Display conditions
1) View_Detail_Plan_Simulation? and (Trim_Op? or User?=="Developer")
Explanation: [✓ constant] equation
Explanation text: http://save.help_pictures/trim_explain.html
Explanation graphics:
Associated Help Pages:
Notes: This is where you place your internal documentation notes

Figure J-5: Cost Model Developer Screen Example

3.0 SAVE Cost Estimating Model Functionality

Four cost models were developed under the SAVE program: sheet metal, assembly, machining, and hand lay-up composites. The intent of these models was to demonstrate the ability of utilizing the simulation data available through the SAVE tools to improve cost estimating accuracy and reliability. The following section describes the underlying cost estimating shell tool used, model descriptions and capabilities, feature based costing description, and typical data elements.

3.1 SAVE Program Cost Estimating Tool

The SAVE cost models are built using Cognition Corporation's Cost Advantage™. The product is a Design for Manufacturing (DFM) expert system shell. It is a knowledge-based software system that provides expert-level design guidance and can analyze manufacturing alternatives and producibility, returning a predictive cost analysis. In essence, it captures manufacturing process knowledge and uses that information to identify cost drivers. It supports evaluation of a design based on features, materials, and processes. The tool has the ability to assign costs to these attributes and provide a total cost estimate of a part or assembly. While SAVE is only calculating cost based on manufacturing constraints, Cost Advantage™ can be used for developing costs for other phases of the life cycle. Cost Advantage™ runs on several Unix-based operating systems, as well as on a PC running the NT operating system.

3.2 SAVE Cost Model Descriptions and Capabilities

The following cost estimating models were developed for the SAVE program. Their capabilities and brief descriptions follow. These models are available through the Cognition Corporation and provide a useful starting point for developing similar cost models. Additional descriptions for customizing these models are included in the SAVE Cost Model Development Guide, SAVE Software Product End Item Description Report, Chapter 6.

3.2.1 Assembly Cost Model Description and Capabilities

3.2.1.1 General Description of the Assembly Cost Model

The assembly cost model is designed around assembly oriented manufacturing operations. These are stored in Cost Advantage™ as CA Features. A cost is calculated using data from the SAVE system, such as the process plan and business data; design data from Costlink; and user inputs from within Cost Advantage™. Additional manufacturing operations and design features with their associated cost estimating relationships can be added by the Cost Advantage™ model developer.

3.2.1.2 Manufacturing Operations Currently in the Assembly Cost Model

Setup	Align	Locate
Drill / Drill Ream	Resistance Spotweld	Back Drill
Bench Drill	Spot Face	Drill Out
Ream	Finish Ream	Seal
Verify	Record	Torque
Inspect	Install	Assemble
Attach	Remove	Shim
Cold Work	Packing	Bond Check
Deburr	Apply	Rivet

3.2.1.3 Cost Related Capabilities of the Assembly Cost Model

Major cost outputs included in the Cost Advantage™ summary window:

- Labor Hours
- Labor Cost Dollars
- Total Recurring Cost Dollars
- Tool Cost Dollars

- Total Non Recurring Cost Dollars

Secondary cost and hours breakouts:

- Manufacturing Theoretical First Unit Hours
- Average Hours for the Operations
- Engineering Total Cost
- Manufacturing Engineering Total Cost
- Sustaining Labor Cost
- Material Related Labor Cost
- Assembly Labor Cost
- Quality Assurance Labor Cost

Additional cost capabilities:

- One- and three-tier learning curve options
- Inflation escalation for then year Dollars
- Default standard hour based equations
    - Can be customized to reflect individual plant capability
- Capability to import simulation hours from SAVE

3.2.1.4 User Related Functionality

The following reflect capabilities that the user might want to enhance in the future to further customize the cost model from both a cost estimating and a display perspective:

- Type of Output (Manual vs. Auto)
    - Facilitates stand-alone usage of the cost models without cluttering the display when working in the SAVE environment.
- Estimate Type (Working, Review, Released)
    - Identifies the state or condition of the SAVE design trade.
- User
    - A function to customize displays and capabilities for different user groups.

- Program
    - This can be customized to reflect the programs worked at the facility.
- Site
    - This variable can be used to specify company locations as well as vendor sites. Use these to customize access into rate tables or to modify cost estimating relationships to reflect the capabilities of a specific site or vendor.
- Aircraft Quantity
    - Customize this variable for the product quantity.
- View Factors
    - These variables are used to control the display of other variables.

3.2.2 Sheet Metal Cost Model Description and Capabilities

3.2.2.1 General Description of the Sheet Metal Cost Model

The Sheet Metal cost model is designed around families of parts and their associations to manufacturing processes and design features. A cost is calculated using data from the SAVE system, design data from Costlink, and user inputs from within Cost Advantage™. Additional manufacturing operations and design features with their associated cost estimating relationships can be added by the Cost Advantage™ developer. Component cost models such as the sheet metal, machining, and composites models have the design features residing in the CA Feature section, and manufacturing operations in the CA Process area.

3.2.2.2 Manufacturing Operations Currently in the Sheet Metal Cost Model

Layout	Shear	Drill
Rout	Hydroform Fluid Cell or Hydraulic Press	Corrosion Protection
Heat Treat	Age Harden	Inspect
Mark	Sand	Trim
Mask	Clean	Deburr
Straighten		

The model is designed around providing the capability to add additional part families as well as additional manufacturing operations, cost estimating relationships, and design features.

3.2.2.3 Major Design Feature Categories Included in Sheet Metal Cost Model

- Openings and Cutouts
    - Round Holes

    - Square Holes
    - Other Holes
- Forming
    - Bends
    - Joggles
    - Flanges
    - Beads
- Contour
- Material
    - Type and Alloy
    - Density
    - Initial and Final Material Condition

3.2.2.4 Cost Related Capabilities of the Sheet Metal Cost Model

The summary cost categories in the sheet metal, machining, and hand lay-up composites cost models were developed on an older version of Cost Advantage™; therefore, they may not use the same conventions as in the assembly model discussed in paragraph 3.2.1.3.

Major cost outputs included in the Cost Advantage™ summary window

- Process Cost
- Material Cost
- Tooling Cost

Secondary cost and hours breakouts:

- Manufacturing Theoretical First Unit Hours
- Average Manufacturing Hours per Component
- Engineering Total Cost
- Manufacturing Engineering Total Cost
- Sustaining Labor Total Cost

- Material Related Labor Cost
- Quality Assurance Labor Cost

Additional cost capabilities:

- Three-tier learning curve
- Inflation escalation for then year dollars
- Default standard hour based equations
    - Can be customized to reflect individual plant capability
    - Reflect setup and run time

3.2.2.5 User Related Functionality

The following reflect capabilities that the user might want to enhance in the future to further customize the cost model from both a cost estimating and a display perspective:

- Type of Input (Manual vs. Costlink)
    - Facilitates stand-alone usage of the cost models without cluttering the display when working in the SAVE environment.
- Estimate Type
    - Identifies the state or condition of the SAVE design trade.
- Part Type
    - This is a subset of the part family. It is used in selecting the appropriate process plan template.
- User
    - A function to customize displays and capabilities for different user groups.
- Program
    - This can be customized to reflect the programs worked at the facility.
- Process Plan from SAVE DB
    - A toggle for condition when a process plan is available from the SAVE manufacturing simulations. If the toggle is off (i.e., there is no plan from SAVE), a template within Cost Advantage™ is used.

- Fabrication Site
    - This variable can be used to specify company locations as well as vendor sites. Use these to customize access into rate tables or to modify cost estimating relationships to reflect the capabilities of a specific site or vendor.
- Aircraft Quantity
    - Customize this variable for the product quantity.
- View Factors
    - These variables are used to control the display of other variables.

3.2.3 Machining and Hand Lay-up Composites Cost Models

The machining and hand lay-up composite cost models were developed with the same design philosophy. They were designed around providing the capability to add additional part families as well as additional manufacturing operations, cost estimating relationships, and design features. Like the sheet metal cost model, they are designed around families of parts and their associations to manufacturing processes and design features. Costs are calculated using data from the SAVE system, design data from Costlink, and user inputs from within Cost Advantage™. Additional manufacturing operations and design features with their associated cost estimating relationships can be added by the Cost Advantage™ developer. Design features reside in the CA Feature section, and manufacturing operations in the CA Process area. Cost Estimating relationships are calculated in external spreadsheets, placed in an ASCII file, and accessed based on rules and equations in the Cost Advantage™ models. To customize these models, the ASCII files may updated with values that reflect the facility operations.

3.2.3.1 Cost Related Capabilities of the Cost Models

The summary cost categories in the sheet metal, machining, and hand lay-up composites cost models were developed on an older version of Cost Advantage™; therefore, they may not use the same conventions as in the assembly model discussed in paragraph 3.2.1.3.

Major cost outputs included in the Cost Advantage™ summary window:

- Process Cost
- Material Cost
- Tooling Cost
- Total Cost

Secondary cost and hours breakouts:

- First unit T1 hours

- Average production unit hours
- Tool manufacturing hours
- Sustaining tool manufacturing hours
- Sustaining tool engineering hours
- Tool material dollars

Additional Cost Capabilities:

- Five-tier learning curve
    - Utilizes external subroutine
- Inflation escalation
- Default standard hour based equations
    - Can be customized to reflect individual plant capability
    - Reflect setup and run time

3.2.3.2 User Related Functionality

The following reflect capabilities that the user might want to enhance in the future to further customize the cost model from both a cost estimating and a display perspective:

- Part Family
    - This is a selection at the top of the Process window.
- Lot Order Quantity
    - Lot size.
- Program Quantity
    - Number of product to be built for the program
- View Factors
    - These variables are used to control the display of other variables.

3.2.3.3 Major Design Feature Categories in the Machining Cost Model

- Pocket
- Holes

    - Cut Out
    - Fastener hole
- End Cut or Angle Cut
- Material
    - Type
    - Billet Thickness
    - Temper
    - Product form

3.2.3.4 Major Design Feature Categories in the Hand Lay-up Composites Cost Model

- Contour
- Internal Drop / Buildup
- Cutout
- Plies
- Material
    - Type and Form

3.2.3.5 Components Included in the Machining Cost Model

- Airframe Structure
    - Skin
    - Cover or Door
    - Rib or Spar
    - Frame
    - Bulkhead
    - Shear Web
    - Leading Edge
    - Longeron or Beam

3.3 Feature Based Costing Overview

The SAVE program utilizes feature-based cost estimating models. These cost models use the relationships between design features and manufacturing processes to provide cost information about the component or assembly. Each part family will have different key cost driving characteristics that are defined by the IPT. A sheet metal part and its features are illustrated in Figure J-6. Many of these part features are common to the composite part illustrated in Figure J-7. When the SAVE cost models were developed, common features were implemented for the machined and hand lay-up composite parts. Lessons learned were implemented in the cost knowledge base.

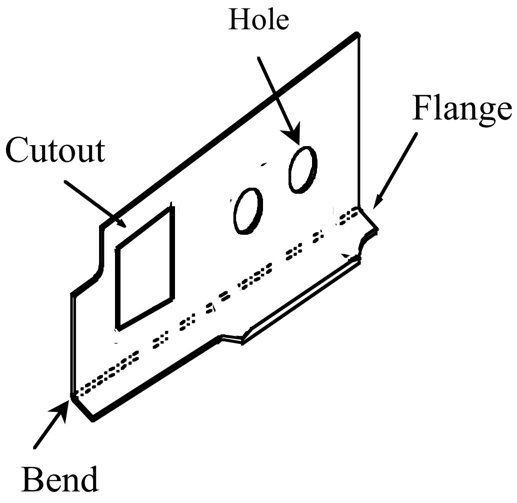


Figure J-6: Cost Driving Features for Machined Part Example

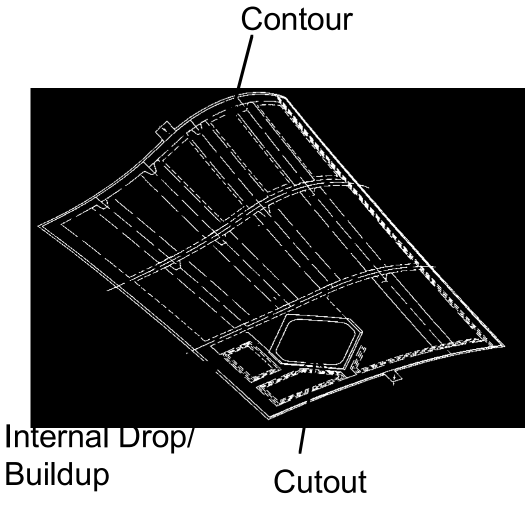


Figure J-7: Example of Hand Lay-up Features

The following are examples of features and part characteristics common to many cost estimating knowledge bases:

- Component length and width
- Component thickness - minimum and maximum
- Hole diameter and tolerance
- Contour
- Material type and form

Additional features that are only found in composites, such as numbers of plies and drops/ buildups, can be written into the composites knowledge base.

3.4 Typical Data Elements in a Cost Model

The SAVE cost estimating tool provides the capability to input and output other important information besides the features described above. Figure J-8 describes typical types of data that are included in the cost models. Learning curve formulas and methods for building up the product cost are built into the SAVE models. These can be customized to reflect a company's particular business environment. Additional types of cost breakdowns can easily be added to the model to support the decision making process. Tables for labor rates, burden factors, and material costs have been developed externally to Cost Advantage™, allowing for easy updates and customization.

Cost Inputs	Cost Outputs
Feature Parameters	Recurring Manufacturing Labor Cost
Material Type	Recurring Material Cost
Process Selection	Non-recurring Tool Cost
Number or Units	Non-recurring Engineering Cost
Units per Aircraft	First Unit Cost
Weight	Tooling Cost
Programmatics	Quality Assurance Cost
Other	Process Plan Simulation

Figure J-8: Typical Cost Model Data

4.0 Integration of Cost Advantage™ with SAVE and Costlink

The uniqueness of the SAVE cost models is their capability to integrate with design and simulation data. This section briefly describes the capabilities of these two functions. More detailed descriptions are included in separate sections of this document.

4.1 Integration between the CAD tool CATIA™ and Cost Advantage™

The CAD tool used to demonstrate SAVE is CATIA™, a 3-dimensional design tool widely used by aerospace companies. It provides part, assembly, tool, inspection equipment, and support equipment designs and data for numerically-controlled (NC) programs. The Costlink software developed by Cognition Corporation for SAVE extracts pertinent design information from CATIA™ and makes it available to the cost estimating session. The data is stored in the Cognition Corporation tool Knowledge Center (KC) and is imported into the cost estimating session in Cost Advantage™. The designer can access Cost Advantage™ from a CATIA™ session, or the cost estimator can access previously saved design data for inclusion in a trade study. See Appendix I for more information.

4.2 Integration between Cost Advantage™ and the SAVE system

For integration between Cost Advantage™ and the SAVE system, a map file is used which correlates the variable names in the cost model with those used in the SAVE database. More information on this topic is available in Appendix E, the Cost Advantage™ Wrapper User's Guide.

5.0 Cost Tool Implementation

There are several facets to implementing a cost estimating tool into an integrated environment such as SAVE. These include identifying product families, understanding their cost driving features, identifying relevant manufacturing processes, and developing associated cost estimating relationships. The developers need to work closely with their ultimate system users and data sources to ensure the best models and end-user buy-in for the system.

The first step towards implementing the SAVE cost estimating system is to work with the cost estimators, designers and manufacturing personnel to identify the components that are most beneficial to include in the system. Next, identify the cost driving features of these part families and relate them to your manufacturing processes. In-depth research is then required to define manufacturing planning performed at the factory, limitations of the equipment, material specifications, time standards, and cost factors.

Once the research is complete, the next phase is the design phase. This encompasses establishing variables and the designating variable locations within the cost tool. Relationships to a SAVE compliant database are also established here. The next phase is to program the variables and cost estimating relationships (CERs) into the cost tool, utilizing templates like those developed under the SAVE program. Once this phase is complete, a validation activity is required to make sure the information is reliable. It is important to include your end users in these activities so that they are comfortable with the features and CER approaches that are selected.

Cost Advantage™ contains three variable categories: material, process, and feature. The cost and design characteristics are allocated into these three areas. Cost estimators or value engineers are typically the ones who will be implementing the cost estimating relationships into this tool. A producibility engineer or manufacturing engineer will provide producibility rule support. It is critical to document the model with comments about the CERs and producibility guidance. Cost Advantage™ provides the capability for the developer to record internal notes regarding each object or formula. Other help information can be documented for access by the end users. This help can be embedded in the cost tool, located in external files, or accessed from the web. The user can easily access this information while working on his cost trade.

Both cultural and political issues need to be considered when implementing an expert system cost model such as the SAVE tool. Agreement is required by all affected departments for this tool to be accepted and utilized. This is a new way to do business for many companies, so this acceptance is critical to the success of the program. This cost tool provides the Integrated Product Team (IPT) a way to rapidly do design trades that include cost. The designer could potentially use this tool on his own, although this should only occur for straightforward trades. The bounds for a designer using this tool with out a cost estimator need to be understood and agreed to by all groups. The ideal situation for using this tool is for the designer and cost estimator to sit together and utilize the SAVE cost tool during their design trade.

The screen previously shown in Figure J-4 is representative of the type of information that the end user would see when utilizing the SAVE cost tool. Both inputs and outputs are readily accessible during the trade study. The user can also query the system for help during his session. When implementing the system, the developer should work with the end users to ensure that the appropriate information is presented on the user's screen.

There are several things that can be done to maximize the benefit and usefulness of the system. First, training is very important for both the users and developers. Secondly, system maintenance is required to avoid the potential problem of data obsolescence. Developing a plan for updating the CERs as the factory and products evolve can accomplish this. This plan should also include a scheme for material costs and labor rate updates.

6.0 Summary

There are many benefits to implementing a cost estimating tool in a manufacturing simulation suite such as the SAVE integrated environment:

- SAVE’s tool suite integration architecture provides a seamless data-exchange environment for incorporating manufacturing simulation information into the cost assessment.
- The integrated cost models in the SAVE architecture provide a consistent, repeatable approach for estimating part costs.
- The cost estimating tool supports discussions between the cost estimator, manufacturing personnel, and the design engineers for making design decisions.
- Potential manufacturing problems are identified early in the design process, reducing scrap and rework costs.

- Integrated cost and design tools provide a better understanding of the cost implications of design features and characteristics. Such tools support cost effective redesigns, as well as the identification of better product/process alternatives.

Appendix K

Microsoft Project Wrapper User's Guide

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

1.0 Introduction

Microsoft Project 98 has the ability to import and export SAVE Process Plans as schedules via a CORBA compliant Visual Basic application which bridges Project 98 to SAVE. The Visual Basic application makes use of Iona's CORBA/ActiveX Bridge to supply the Project 98 to SAVE link.

The Project 98 Wrapper provides a graphical user interface into the SAVE library, from which the user may create or select Simulation Request, Design Studies, Design Study Alternatives, and Process Plans. The end results are that a Project 98 schedule is exported into SAVE as a Process Plan, or a SAVE Process Plan is imported as a schedule into Project 98. The User interface is window driven, which guides the user through the selection process of exporting or importing.

2.0 Starting the Project 98 Wrapper

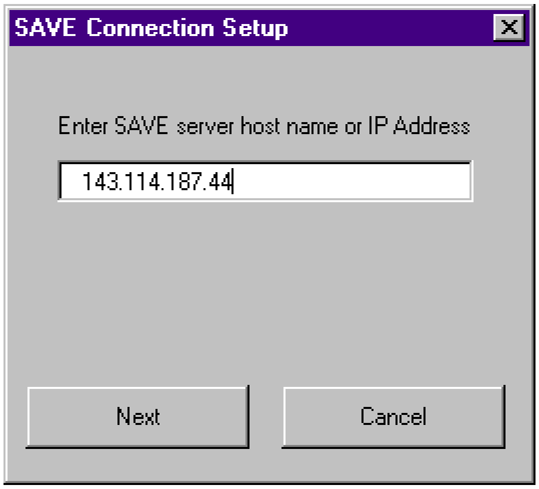
The Project 98 Wrapper takes only one argument. To start the Wrapper, click on the SAVE_Database toolbar button displayed at the top of the Microsoft Project 98 Gantt Chart window. Upon installation, this toolbar should have been configured with the server argument that specified the IP address and name for the SAVE server.

3.0 Wrapper Interface

The Project 98 Wrapper interface consists of a series of windows which prompts the user for input, advises the user of possible selections and displays status as progress is made. The primary windows are The CORBA Connection window, Initial Selection window, Simulation Request window, Import Design Study Process Plan window, Export Process Plan to SAVE window, and Process Plan Progress window.

3.1 CORBA Connection Window

The CORBA Connection window (Figure K-1) prompts the user to enter the IP address or machine name of the machine hosting the SAVE server. A default IP address is provided at installation, but the user may override this address if desired. When the user selects next, an attempt to connect to the SAVE server is made. If the connection is successful, the Initial Selection window is displayed.


SAVE Connection Setup

Enter SAVE server host name or IP Address

143.114.187.44

Figure K-1: CORBA Connection Window

3.2 Initial Selection Window

Figure K-2 shows the Initial Selection window. The user will have several options to choose from. The Import Parts List selection is intended to allow the user to import a BOM (Bill of Materials) in the form of a text file, which will be displayed as a set of tasks in MS Project. At this time, the Import Parts List option is not functional. The second option starts the process, which allows the user to select and open a Simulation Request. The third option starts the process, which allows the user to select and open a Design Study Process Plan. Finally, the fourth option starts the process, which allows the user to export an MS Project schedule to SAVE. If the Gantt chart from which the wrapper is started is blank, the export option will not be enabled. Once an option is selected, the user presses <Next> to continue. The user may also select <Cancel> to exit the Wrapper.

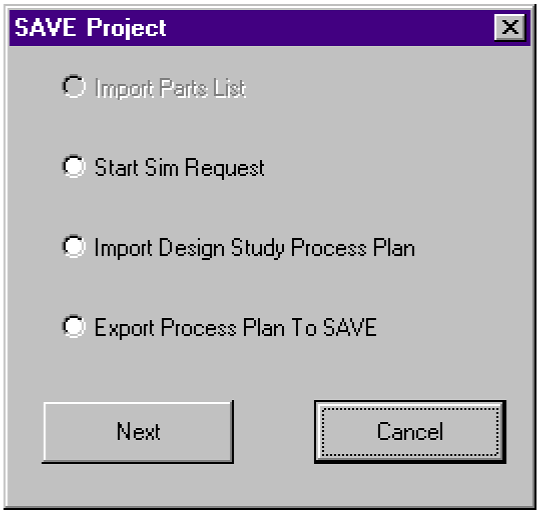
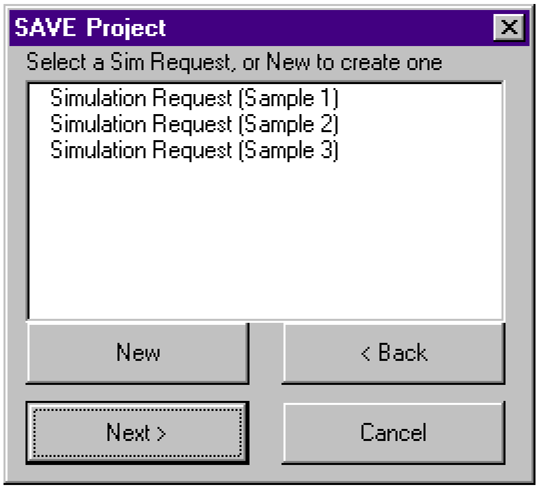

SAVE Project
☒
○ Import Parts List
○ Start Sim Request
○ Import Design Study Process Plan
○ Export Process Plan To SAVE
Next
Cancel

Figure K-2: Initial Selection Window

3.3 Simulation Request Window



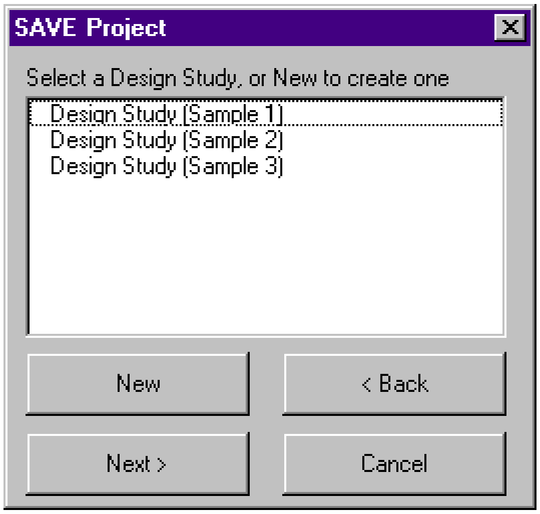
Select a Sim Request, or New to create one

Simulation Request (Sample 1)
Simulation Request (Sample 2)
Simulation Request (Sample 3)

Figure K-3: Simulation Request Window

The Simulation Request Window (Figure K-3) will display a list of Simulation Request found in the SAVE library. The user may also select <New>, which will prompt the user, for the name and description of a new Simulation Request. If the SAVE library does not contain any Simulation Request, the user will be immediately prompted to enter the name and description of a new Simulation Request. From the Simulation Request window, The user may also select <Back>, which returns to the Initial Selection Window, or <Cancel> to exit the Wrapper. If the user makes a selection and presses <Next>, the Process Plan Progress Window will appear. The Process Plan to be displayed (as a Project schedule) will always be from the Simulation Request ProcessPlan object. The only way to view an alternative process plan, is to open a Design Study via the Import Design Study Process Plan option on the Initial Selection window.

3.4 Design Study Window


SAVE Project

Select a Design Study, or New to create one

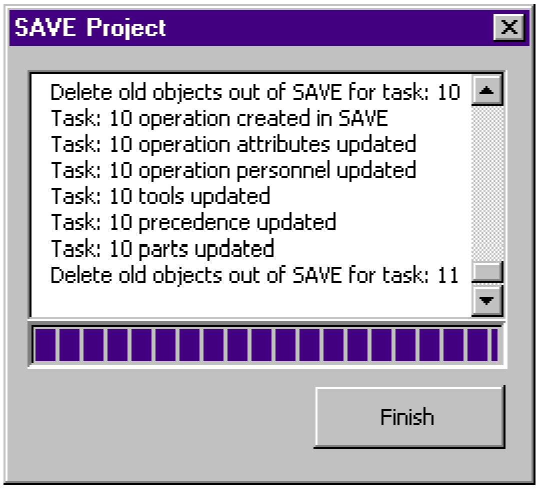
Design Study (Sample 1)
Design Study (Sample 2)
Design Study (Sample 3)

Figure K-4: Design Study Window

The Design Study window (Figure K-4) appears when the user selects the Import Design Study Process Plan option from the Initial Selection window. A list of Design Studies is shown, from which the user may select one. The user may also press <New>, which brings up a prompt for the name and description of a new Design Study. If the SAVE library does not contain any Design Studies, the user is immediately prompted for the name and description to create a new Design Study. At any time, the user may select <Back> to return to the Initial Selection window, or <Cancel> to exit the Wrapper. Once the user has made a selection and pressed <Next>, a Design Study Alternative must be selected. If the Design Study has both a Selected Alternative and a list of Alternatives, the user will be prompted to choose either the Selected Alternative, or to view the list of other Alternatives. If only the Selected Alternative object exist, then the list of process plans contained within it are listed, so that the user can select one. Similarly, if only the Alternative lists object exist, the list of alternatives are displayed for the user to select. When the user selects an alternative from the list, the process plans within that object are listed. Once the user selects a process plan from the list (either from the Selected Alternative or an alternative from the Alternative list), the Process Plan Progress window is displayed to show progress as the process plan is imported into MS Project. Although the user can select and display a process plan from the Selected Alternative object, the Wrapper will not allow the user to export a Project Schedule to a Selected Alternative. All Project schedules are saved to process plans owned by Alternatives from the Alternative list. The Selected Alternative will point to one of the alternatives from the Alternative list once the team has made a final design selection from the list of Alternatives. Assigning the final selection to the Selected Alternative is accomplished with the Query Manager tool.

3.5 Process Plan Progress Window

The Process Plan Progress window (Figure K-5) displays status messages as a process plan is either imported or exported. The finish button is disabled until the import or export is complete.


SAVE Project

```
Delete old objects out of SAVE for task: 10
Task: 10 operation created in SAVE
Task: 10 operation attributes updated
Task: 10 operation personnel updated
Task: 10 tools updated
Task: 10 precedence updated
Task: 10 parts updated
Delete old objects out of SAVE for task: 11
```

Figure K-5: Process Plan Progress Window

4.0 Wrapper Installation

Step 1:

Under the Install Kit folder, open the CORBA COM SAVE Bridge folder and copy the files to a folder on your machine.

Step 2:

If the Orbix runtime is not loaded on your machine, you will need to install c:\IONA\Orbix_2.3c\BIN\ioleM23C.dll

Step 3:

Locate REGSVR32.exe on your machine and Note its path

On the Start menu click Run

In the dialog box type the following:

```
<path to RegSvr32>\RegSvr32.exe C:\IONA\Orbix_2.3c\Bin\iolem23c.dll
```

On the Start menu click Run

In the dialog box type the following:

```
<path to RegSvr32>\RegSvr32.exe <path to CORBA COM SAVE Bridge
files>\SaveBroker.dll
```

Step 4:

Before installing Project 98, run a search on your machine for COMCTL32.OCX. It is generally found at C:\WINNT\system32 If found proceed to Step 5. If not found, copy the version in the ActiveX Control folder to C:\WINNT\system32. You must now register this control.

On the Start menu, click Run.

In the Run dialog box, type the following:

```
<Path to RegSvr32>\REGSVR32.EXE  <Path to OCX>\OCXFILE.OCX
```

For example:

```
C:\Devstudio\VB\REGSVR32.EXE C:\Winnt\System32\COMCTL32.OCX
```

NOTE: If Regsvr32.exe is in the System or System32 folder, the path is optional.

Step 5:

a) Install Project 98. Once installed, start it.
b) From the menu bar, select Tools -> MACRO -> Visual Basic Editor.
c) From the Visual Basic menu bar, select Tools -> References...

d) In the Reference window, select Standard Orbix Types. If you can't find it, select Browse then go to: c:\IONA\Orbix_2.3c\BIN\ioleM23C.dll and open the dll. It should now be selected in the Reference window.

e) Select the Browse button again, and go to the folder where you copied the CORBA COM SAVE Bridge files. Open SAVEBroker.DLL. SAVEBroker Type Library should now be checked in the Reference window. Click OK to close the Reference Dialog Box.

f) Close the Visual Basic Editor

Step 6:

a) From the Project 98 menu bar, select File -> Open. Go to the Install Kit\The Wrapper folder, and open Wrapper.mpp

b) From the Project 98 menu bar select Tools -> Organizer...

c) In the organizer box, select the Modules tab. Select all the modules in the Wrapper window, and move them to the Global.mpt window.

d) Select the Toolbars tab in the Organizer box. Copy the Save toolbar to the Global.mpt window.

e) Select the Tables tab in the Organizer box. Move the SaveDb table to the Global.mpt window. Close the Organizer box.

Step 7)

You are now ready to run the Wrapper. Make certain that a SAVE Server is running, and that you know the IP address of the server. Select the SAVE Database toolbar button, and follow the prompts.

5.0 Wrapper Software Design

The MS Project Wrapper is a Visual Basic application, which makes use of Iona's ActiveX to CORBA Bridge. The SAVE IDL is compiled using Iona's winidl compiler, which creates visual basic objects for each SAVE CORBA object. The Visual Basic application is embedded into MS Project 98 as a macro. The application is divided into two major components. The form components are the graphical user interfaces, while the module components provide the underlying SAVE to MS Project conversions.

5.1 User Interface Forms

The wrapper makes use of the following graphical user interface forms.

5.1.1 frmSAVEConnect

This form prompts the user for the IP address used to connect to the SAVE server.

5.1.2 frmSAVEErrorBox

This form displays an error message, that too many Comm Failures are occurring. The User is prompted to cancel the wrapper execution, or to continue trying.

5.1.3 frmSAVEexport

This form prompts the user on export choices, based on the control state the wrapper is in.

5.1.4 frmSAVEimportDs

This form, prompts the user for any Design Study inputs needed.

5.1.5 frmSAVEinit

This form is the initial page that prompts the user to choose between importing a part list, or opening a simulation request, or opening a design study, or exporting a MS Project schedule to SAVE.

5.1.6 frmSAVEpartsList

This form prompts the user to import a parts list. For now, this code is not used, waiting for requirements.

5.1.7 frmSAVEprogress

This form is a progress window, which displays progress when a SAVE process plan is imported into MS Project or a schedule is exported to SAVE.

5.1.8 frmSAVESimReq

This form prompts the user for Simulation Request inputs.

5.2 Wrapper Modules

The Wrapper makes use of the following modules to translate SAVE objects to MS Project data, and vise versa.

5.2.1 SAVE_Database

The SAVE_Database module provides methods to connect and disconnect from the SAVE server. Methods are also provided to retrieve library list objects from the SAVE library, as well as the list item count and list item names.

5.2.2 SAVE_DesignStudy

The SAVE_DesignStudy module provides methods for accessing Design Study objects as well as objects within a Design Study object. The following is a breakdown of the subroutines and functions provided by this module.

5.2.2.1 CreateDesignStudy

This subroutine creates a new Design Study in SAVE, and assigns the module pointer DesignStudy to it. Next the Design Study's Alternative object is retrieved from SAVE, the module pointer DsAlternative is set to this object.

5.2.2.2 WrDsSaBpp

This subroutine creates and fills in the Design Study Selected Alternative Baseline Process Plan. First, a process plan is created using the name and description found in task(1) of the Project file. Once the Process Plan is created in SAVE, a pointer (SAVE_ProcPlan.ProcPlan) is assigned the new Process Plan. Calling SAVE_ProcPlan.WriteProcPlan fills in the SAVE version of the Process Plan.

5.2.2.3 GetDesignStudy

This subroutine retrieves a Design Study from the SAVE library based on the index passed as a parameter. The module pointer DesignStudy is set to the object retrieved. The Design Study's Alternative list object is then retrieved, and the module pointer DsAlternative is set to this object. A flag is then set to indicate whether the list is empty or not.

Next, the Design Study's SelectedAlternative object is retrieved. This object is tested, and a flag is set to indicate if the object is null. If it is not, the SelectedAlt module pointer is set to it. The SelectedAlternative's Process Plan list object is then retrieved. A flag is set to indicate if the list is empty.

5.2.2.4 GetDsByName

This function retrieves a Design Study from the SAVE library using the name specified in the public variable DesignStudyName. To do this, the function gets each Design Study one by one from the SAVE library, until a name match is found. Once the object is found, the module pointer DesignStudy is set to the object.

5.2.2.5 GetDsAlt

This subroutine retrieves a Design Study Alternative from the Design Study Alternative list based on the index passed.

5.2.2.6 SetDsAlt

This subroutine sets the module pointer DesignStudyAlt to the Design Alternative object passed as a parameter. The Design Alternative Process Plan list is retrieved, and the module pointer ProcPlanSeq is set to it.

5.2.2.7 GetDsAltByName

This function searches for the Design Study Alternative by name, using the name specified in the public variable SelectedDesignStudy. Typically, the caller sets this variable. Once the Alternative is found, the object is retrieved and the module pointer DesignStudyAlt is set to it. The Alternative's Process Plan list is retrieved, and the module pointer ProcPlanSeq is set to it.

5.2.2.8 ReadBaselineProcPlan

This subroutine retrieves the BaselineProcPlan object and calls SAVE_ProcPlan to have it displayed.

5.2.2.9 ReadPpsPp

This subroutine gets a Process Plan object from the list of Process Plans, and calls SAVE_ProcPlan to display the Process Plan as an MS Project schedule.

5.2.2.10 AddDsAlt

This subroutine creates a new Design Alternative using the name found in the public module variable SelectedDesignStudy, and the description found in the public module variable DesignStudyDesc. Once the object is created, the pointer DesignStudyAlt is set to it. Finally the new Design Study Alternative is added to the list of Alternatives.

5.2.2.11 OwrBaselinePp

This subroutine gets the BaselineProcPlan object; sets the pointer SAVE_ProcPlan.ProcPlan to it, and then calls SAVE_ProcPlan to overwrite this object with the data found in the MS Project schedule.

5.2.2.12 WrDsAltBpp

This subroutine creates a new Process Plan, sets the SAVE_ProcPlan.ProcPlan pointer to it, and calls SAVE_ProcPlan to fill it in with the MS Project schedule data. Once finished, the DesignStudyAlt.BaselineProcPlan pointer is set to the filled in process plan.

5.2.2.13 AddDsAltPpsPp

This subroutine adds the process plan pointed to by the SAVE_ProcPlan.ProcPlan pointer to the list of Process Plans pointed to by the module pointer ProcPlanSeq.

5.2.2.14 OwrDsPpsPp

This subroutine gets the indexed Process Plan object from the list of Process Plans. Sets the SAVE_ProcPlan.ProcPlan pointer to it, and then calls SAVE_ProcPlan to overwrite it with new data from the MS Project schedule.

5.2.2.15 CreateAddPpsPP

This subroutine creates a new Process Plan, sets the pointer SAVE_ProcPlan.ProcPlan to it, fills it in with data from the MS Project schedule, and finally adds the filled in process plan to the list of process plans pointed to by the module pointer ProcPlanSeq.

5.2.2.16 GetAltSeqCount

This function returns the number of Design Study alternatives listed in the Design Study Alternative list.

5.2.2.17 GetDsAltName

This function returns the name of an indexed Design Study object from the list of Design Study objects.

5.2.2.18 GetSelBlPpName

This Function returns the name of the selected Baseline Process Plan. The SelectedAlt module pointer is used to retrieve the BaselineProcPlan object. The name of the BaselineProcPlan object is returned.

5.2.2.19 GetAltBlPpName

This function returns the name of the BaselineProcPlan object pointed to by the DesignStudyAlt module pointer. The DesignStudyAlt pointer points to a Design Alternative from the Design Alternative list.

5.2.2.20 GetPpCount

This function returns the number of Process Plans listed in the list of Process Plans.

5.2.2.21 GetPpName

This function returns the name of an indexed Process Plan from the list of process plans.

5.2.2.22 SetDesignStudy

This subroutine takes the Design Study object passed to it, and sets the module pointer DesignStudy to it. The DesignStudy Alternative list object is then retrieved and the module pointer DsAlternatives is set to it. A flag is set to indicate if the list is empty. Next the DesignStudy SelectedAlternative object is retrieved. A flag is set to indicate if the object is empty. If the Object is not empty, the SelectedAlternative Process Plan list object is retrieved. A flag is set to indicate if this list is empty.

5.2.3 SAVE_ErrorHandler

The SAVE_ErrorHandler provides error-handling capability. Visual Basic does not have Try/Catch statements, which you would normally use when executing CORBA commands. Instead, each method which has a CORBA command, has a ON ERROR GOTO ERRORHANDLER statement. Each ERRORHANDLER in turn calls this Function. This function will examine the Err.Number to determine if the error is CORBA related. If the error is CORBA, then function GetCorbaError is called to return a string description of the error. Individual case statements handle the exceptions that the SAVE server throws. All other CORBA errors are handle by a Case-Else statement.

All errors are fatal, with one exception. CORBA COMM_FAILURE is not fatal. This error generally occurs when a CORBA command times out before completing. In this case this function returns a true flag, to indicate the CORBA command should be tried again.

COMM_FAILURE requires special handling. We do not want to get into an infinite loop re-trying. A timer is used to time how far apart the COMM_FAILURE are occurring. Also a counter keeps track of the total number of COMM_FAILURE. In the event that a COMM_FAILURE occurs after both read and write transactions to the server are closed, then the failure occurred during the closing of the read transaction. In this case we should just exit, and not attempt to re-close the read transaction.

5.2.4 SAVE_Globals

This module holds all the global variables, which are used by the other SAVE modules.

5.2.5 SAVE_Main

This module is the main starting point for the SAVE wrapper, and the final ending point once the wrapper is complete.

5.2.6 SAVE_ProcPlan

The SAVE_ProcPlan module provides methods for accessing Process Plan objects as well as objects within a Process Plan object. The following is a breakdown of the subroutines and functions provided by this module.

5.2.6.1 DisplayBlankPP

If a user starts the SAVE wrapper from a blank project file, and creates a Sim Request or Design Study, this subroutine will add the new Process Plan name and description to the first task line of Project.

5.2.6.2 DisplayProcPlan

This subroutine is called to display a Process Plan retrieved from the SAVE server. The calling program must set the public module variable ProcPlan to the Process Plan the SAVE server retrieves prior to calling this subroutine.

5.2.6.3 AddTask

The AddTask subroutine creates a task in Project for each Operation within a Process Plan. All relevant Project tasks data is mapped to it's corresponding SAVE Operation attributes.

5.2.6.4 ConvertSaveDate

This function is used to convert the SAVE date format to a format recognizable by MS Project.

5.2.6.5 ConvertProjectDate

The function is used to convert the Project date into a format that is recognizable by SAVE.

5.2.6.6 WriteProcPlan

This subroutine is used to write the MS Project schedule to SAVE. Care must be taken to not destroy data that already exist in SAVE, but that is not used by MS Project. For this reason, if an Operation already exists in SAVE, we do not overwrite it, we only overwrite the fields in it that are used by MS Project.

5.2.6.7 WriteTask

WriteTask is the subroutine that writes each tasks to a corresponding operation in SAVE.

5.2.6.8 GetCount

This function is a utility that determines how many sub strings are in a string that is delimited by the character passed as the first parameter (delim).

5.2.6.9 FillArray

This function is a utility that takes the sub strings in a string, and places them in an array. The size of the array is determined by the parameter "num". The delimiter character is passed in "delim".

5.2.6.10 FindPreds

This function is a utility that takes the Predecessor string from a Project task, and extracts each predecessor, placing it in an array. Since it is difficult to determine how many predecessors are in the string, before hand, we assume no more than 20 will be allowed.

5.2.6.11 AddPred

This subroutine adds predecessor numbers to a given task's predecessor field. If the field is empty, simply add the new number to the field. If the field already has numbers in it, then before adding the next number, add a "," as a delimiter.

5.2.6.12 BuildToolArray

This function fills an array with the tools listed in the string passed. The function returns the number of tools listed in the array. The tool names are placed in ToolArray2. ToolArray2 is a 2-diminsional array. The first dimension holds the name of each tool, while the second dimension holds the quantity of the tool used.

5.2.7 SAVE_SimReq

The SAVE_SimReq module provides methods for accessing Simulation Request objects as well as objects within a Simulation Request object. The following is a breakdown of the subroutines and functions provided by this module.

5.2.7.1 GetSimReq

This function retrieves a Simulation Request from the SAVE database, and stores the Simulation Request object in the module pointer SimReq.

5.2.7.2 WriteProcPlan

This subroutine creates a process plan in the SAVE database, and calls SAVE_ProcPlan to fill the process plan. Once the plan is filled, the SAVE Simulation Request Process Plan is set equal to the new process plan.

5.2.7.3 OwrProcPlan

This subroutine overwrites an existing Process Plan. The index parameter indicates the type overwrite. If Index is -1 then overwrite to the Simulation Request’s active process plan. If Index is 0 then overwrite Simulation Request Alternative Baseline Process Plan. If Index is greater than 0 then get the process plan pointed to by the index from the Design Study Alternate Process Plan list, and overwrite it.

5.2.7.4 CreateSimReq

This subroutine creates a new Simulation request from the name found in the public module variable SimReqName, and description found in the public module variable SimRegDesc. The new object is stored in the public module variable SimReq.

5.2.7.5 ReadProcPlan

This subroutine reads the Simulation Request Process Plan from SAVE, and calls SAVE_ProcPlan to display it in the active Project Worksheet.

5.2.7.6 SetNewSimDs

This subroutine sets the pointer in the Simulation Request, to point to a new Design Study. The Design Study name is held by the SAVE_DesignStudy module, and was put there by the user interface form: frmSAVESimReq. The frmSAVESimReq prompted the user for the name.

5.2.7.7 SetNewDsAlt

This function sets the Design Study Alternative in the Simulation Request. The module variables SimReqName and SimReqDesc are set via user input by the calling frmSaveSimReq.

5.2.7.8 SetNewSimPp

This subroutine creates a new process plan using the name and description found in the module variables SimReqProcName and SimReqProcDesc. These were set by the form frmSAVESimReq based on user input. Once the Process Plan has been created, calling WriteProcPlan fills it in, but only if the Project file actually has Process Plan data. Otherwise the Process Plan is left empty, and a one line Project File is created indicating the name of the Process Plan now in SAVE.

5.2.7.9 GetDesignStudy

This subroutine calls SAVE_DesignStudy to retrieve the Design Study object's Alternate list, and Selected Alternate. The Design Study object is passed as a parameter. Flags are set to indicate if the Alternate List or Selected Alternate exists.

5.2.7.10 SetDsAlt

This subroutine calls SAVE_DesignStudy to retrieve a Design Study Alternative from the Design Study Alternative list based on the index passed. The list of Process Plans from the Alternative is retrieved. A flag is set to indicate if the process plan list is empty. The Simulation Request's Design Alternative is set equal to the Alternative just retrieved.

5.2.7.11 GetDsAlt

This subroutine passes a SimReq.DesignAlternative object from SAVE to the module SAVE_DesignStudy which in turn retrieves the Process Plans list from this Alternative object. A flag is set to indicate if the list is empty.

5.2.7.12 GetProcPlan

This subroutine retrieves the Simulation Request Process Plan, and sets the module pointer SimProcessPlan to the process plan object.

5.2.7.13 GetActivePpName

This function returns the name of the Simulation Request Process Plan.

5.2.7.14 GetBlPpName

This function returns the name of the Simulation Request Design Alternative Baseline Process Plan.

Appendix L

Work Flow Manager User’s Guide

SAVE Software User’s Manual
Contract Number F33615-95-C-5538
CDRL A012

1.0 Introduction

The SAVE infrastructure relies on the Work Flow Manager (WFM) to provide structure and guidance for the process of conducting a design or trade study. The IPPT will use the WFM to layout the sequence of events in the study and to execute those events through communication with the appropriate SAVE tools or users.

The WFM is a Java/CORBA application for creating and running work flow models. A work flow consists of process, task and activity nodes linked together in a hierarchical structure. In addition, the WFM supports dependencies between nodes, which controls how the model runs. Nodes may execute concurrently or sequentially, depending on how the dependencies are developed.

The WFM uses a point-and-click interface to create the models. The nodes are selected from a menubar or toolbar, placed on a layout palette and linked together. Selecting a node for editing displays an editor dialog box for changing the information in a node. The Activity node editor allows setting the name, host, properties and operations for a remote simulator. Simulators provide the interface between the commercial simulation tools and the WFM.

Once the work flow model is complete, the WFM runs the model by issuing commands to the nodes. The color of each node changes to indicate the state of the node at any one time. The work flow may be paused, resumed, terminated and reset for another run with the click of a button.

Finally, a work flow model may be saved to and restored from disk. One work flow model may be imported into another model or a whole new model may be loaded from disk. Models can also be saved to different file names, thus allowing a building block approach to creating work flow models.

2.0 Starting the Work Flow Manager

The WFM takes two arguments. To start the WFM, enter the following command at the DOS prompt:

```
java save.wfm.WFM servers emails [models] &
```

where

- servers is the path and file name for the list of available servers,
- emails is the path and file name of email addresses for WFM users, and
- optional models is the default directory for storing the work flow models.

The server data file "wfm.data" contains the list of available simulator servers by their registered name and host machine. This file must be edited for each installation. It contains a single line entry for each server in the system with the following format:

```
simulator-name host-string
```

where

- simulator-name is the server's registered ORB name with no spaces and
- the host-string is the name or IP address of the server's host computer.

The user may also create multiple server data files and specify each of them on the WFM command line.

The emails file "email.data" contains a list of email addresses that will appear in the drop-down menu in the activity template. This list should contain the e-mail addresses for the users of the interactive simulation tools and will be used to send e-mail notification to that user when the work flow progresses to that point.

As an alternative to the command line execution, the system administrator may create a batch file with the default arguments for a given WFM installation. This batch file may be stored on the user's desktop to allow mouse-click execution.

3.0 Work Flow Manager Interface

The WFM interface consists of a menubar, toolbar, layout area and status bar. Figure L-1 shows this interface.

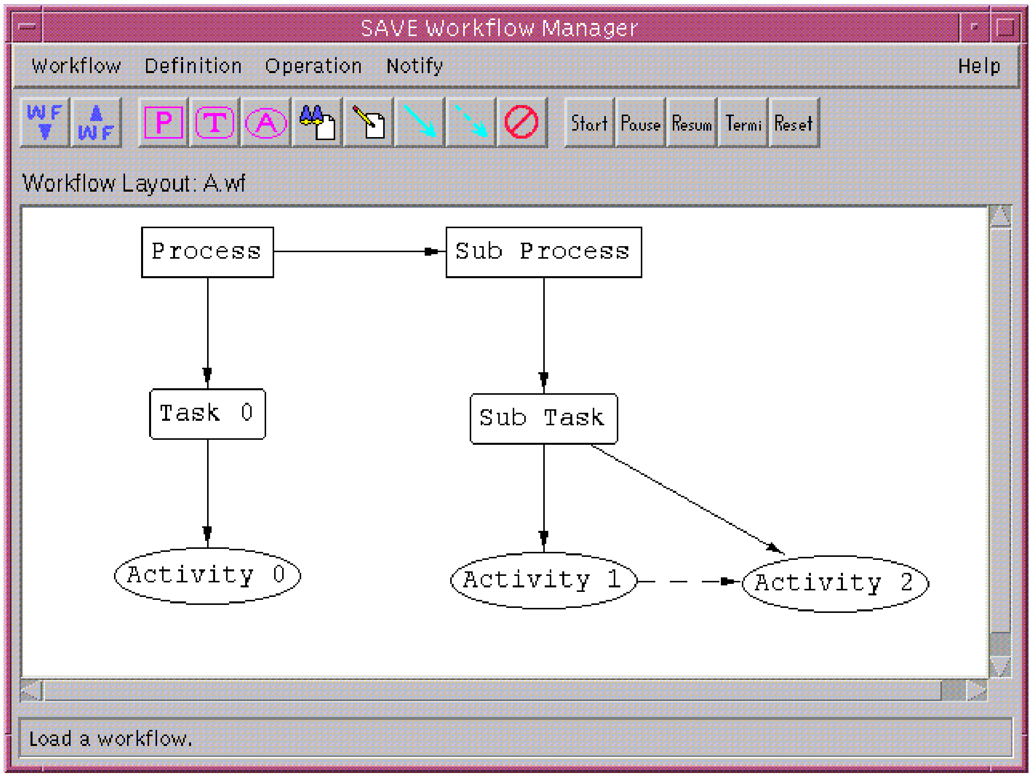


Figure L-1: Work Flow Manager User Interface

The menubar is along the top of the window with a toolbar that implements the primary user tasks just below it. The layout area is in the center of the window and has a label just above it that shows the current work flow model (in this case a model called "`A.wf`"). Under the layout area is the status bar, which displays both the toolbar tips and any messages and usage prompts. The status bar in the figure shows the toolbar tip for the load work flow icon.

3.1 Menubar

The menubar controls the WFM and consists of five primary selections: work flow, definition, operation, notify, and help. These selections are shown in Figure L-2. To select a menubar item, click on the top-level selection and then the desired item. To cancel the operation, just move the cursor off the menubar and click. Selections that are not yet implemented are identified as inactive in the descriptions that follow.

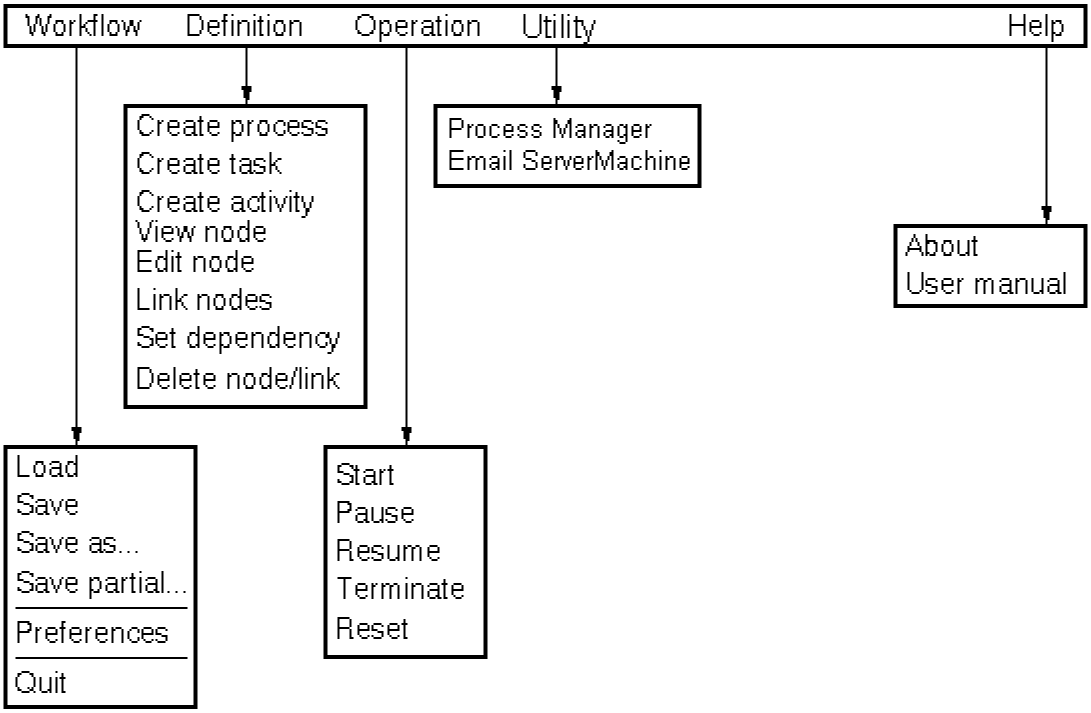


Figure L-2 Work Flow Manager Menubar

3.1.1 Work Flow

The work flow menu item contains selections that allow the user to load or save a particular work flow model. The selections and their definitions are as follows:

- **Load** Load a work flow model.
- **Save** Save the current work flow model using the same name.
- **Save as...** Save the current work flow model using a new name.
- **Save partial...** *(inactive)* Save a selected part of the current work flow.
- **Preferences** *(inactive)* Set any user preferences.
- **Quit** Quit the application.

3.1.2 Definition

The definition menu item contains selections that allow the user to create a work flow model with its nodes and dependencies. The selections and their definitions are as follows:

- **Create process** Create a new process node and place in the work flow model.
- **Create task** Create a new task node and place in the work flow model.
- **Create activity** Create a new activity node and place in the work flow model.
- **View node** Bring up the appropriate node viewer.
- **Edit node** Bring up the appropriate node editor.
- **Link nodes** Make a hierarchical link between model nodes.
- **Set dependency** Make one node dependent on another.
- **Delete node/link** Delete a model node or link.

3.1.3 Operation

The operation menu item contains selections that allow the user to execute the work flow model. The selections and their definitions are as follows:

- **Start** Send the start (prepare and launch) operations to the selected node.
- **Pause** Send the pause operation to the selected node.
- **Resume** Send the resume operation to the selected node.
- **Terminate** Send the terminate operation to the selected node.
- **Reset** Reset the states in the WFM to undefined so a model may be run again.

3.1.4 Notify

The notify menu item contains selections that allow the user to identify the person responsible for the work flow and to establish a mechanism to send e-mail notifications to persons responsible for individual tasks and processes. If tools are interactive and e-mail notifications are desired, the process manager must be defined. Individuals responsible for the tool executions are selected from a pull-down menu in the Activity window. The selections and their definitions are as follows:

- **Process manager** E-mail address of the process manager for the work flow.
- **Email Server Machine** Machine name of a valid e-mail server machine.

3.1.5 Help

The help menu item contains selections that allow the user to obtain information about the use of the work flow manager application. The selections and their definitions are as follows:

- **About** Display the about Work Flow Manager message.
- **User manual** *(inactive)* Display the user manual.

3.2 Toolbar

The toolbar contains icons that represent the major operations on the menubar. They are separated into three groups that correspond to three top-level menubar items. The groups and icons from left to right are:

- **Work Flow**
  - **Load**
  - **Save**
- **Definition**
  - **Create process**
  - **Create task**
  - **Create activity**
  - **View node**
  - **Edit node**
  - **Link nodes**
  - **Set dependency**
  - **Delete node/link**
- **Operation**
  - **Start**
  - **Pause**
  - **Resume**
  - **Terminate**
  - **Reset**

Passing the cursor over a toolbar icon displays its tool tip in the status bar, which provides information about its identity. Moving the cursor off the toolbar icon restores the message that was previously in the status bar.

The way a toolbar icon acts depends upon its function. For those that perform their operation in one step, the icon works like a button. Most of the toolbar icons, however, perform multiple step operations. These toolbar icons "freeze" in the down position and the next step in the sequence appears in the status bar. To cancel one of these multiple step operations, click the toolbar icon a second time.

3.3 Layout

The layout area contains the visual representation of the work flow model. The work flow model in Figure L-1 shows two processes, two tasks and three activities. The processes display as rectangles, the tasks as rounded rectangles and the activities as ovals. The name of each node appears inside the objects. The nodes are all white, which means they are waiting to execute.

The solid lines indicate the hierarchy of the model. For instance, *Process* controls *Task 0* and *Sub Process* while *Task 0* controls *Activity 0*. The dashed line indicates a dependency between *Activity 1* and *Activity 2*. This means that *Activity 1* must complete before *Activity 2* can start. Therefore, in the model shown, *Activity 0* and *Activity 1* will execute in parallel and *Activity 2* will start after *Activity 1* completes.

There are three types of mouse interactions on the layout; selection, movement and viewer. Selection occurs when placing a new node on the layout, linking nodes or deleting a node or link. To make a selection, place the mouse in the appropriate place and click the left mouse button once without moving the mouse.

To move a node on the layout, place the cursor over the node, press and hold the left mouse button and drag the node to its new location. Any links attached to the node move with it. To drop the node, release the mouse button.

The third interaction is a shortcut method for bringing up the data viewer for a node. Place the cursor over a node and click the left mouse button twice in succession without moving the mouse. This brings up the viewer for that type of node. Of course, the user may always select the viewer icon from the toolbar and select the node with a single click.

3.4 Status

The status bar is at the bottom of the display and shows tool tips, messages and usage prompts. A tool tip shows when the cursor passes over a toolbar icon. The messages pop up as determined by the system. Error messages always beep when they display to draw attention to themselves. Usage prompts appear during the multiple step operations such as creating a new work flow node.

4.0 Creating a Work Flow

Creating or modifying a work flow consists of laying down nodes and linking them together. There are three types of nodes (process, task and activity) and two types of links (model and dependency). The combination of the nodes and links create the work flow model.

To create a work flow node, select the type of node from the menubar or toolbar. A prompt appears in the status bar telling the user to place the node on the layout. Click on the layout to drop the node.

To link two nodes together, select the type of link and follow the status bar prompts to select the two nodes. For a model link, select the predecessor node first and then the successor node. For a dependency link, pick the source node first followed by the dependent node. The type of line indicates the type of link, solid for a model link and dashed for a dependency link.

Deleting a node or link is as simple as selecting the delete menubar item or toolbar icon and clicking on the node or link. In the case of a node, the WFM will prompt you first to be sure you want to delete the node. There is no undo operation in the WFM so use care when removing nodes.

Once a node is created, the user may edit it by selecting edit from the menubar or toolbar and clicking on the node. This action displays an editor for that node. A viewer, on the other hand, looks the same as an editor except that it does not allow changes to the data. As a shortcut, a viewer may also be displayed by double clicking on a node.

4.1 Process Template

A process is the top-level node for a work flow model. Processes may contain other processes and tasks, but they cannot contain activities. A process may only have one predecessor node but supports multiple successor nodes. Figure L-3 shows the process template.

The process editor contains entries for the process's name, a one-line description and a multiple line rationale. There is also a successor tree that expands to show all of the successor nodes currently connected to the process. The nodes in the tree cannot be modified directly. To add or remove successor nodes, use the layout to create and delete links. These changes are immediately reflected in the successor node tree.

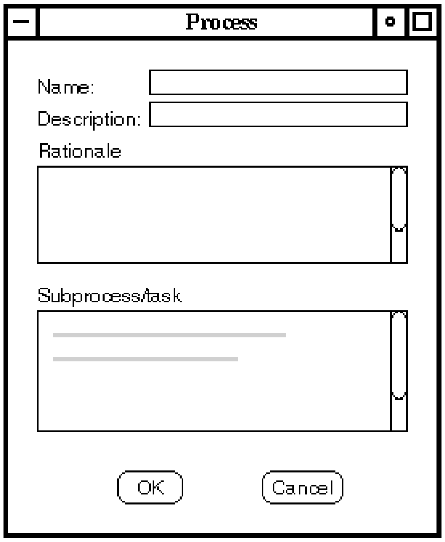


Figure L-3: Work Flow Manager Process Template

4.2 Task Template

A task node serves as the container for activities. Tasks may only contain activities as successor nodes. A task can only have one predecessor node, which must be a process. Figure L-4 shows the task template.

The task editor contains entries for the task's name, a one line description and a multiple line rationale. There is also a successor tree that expands to show all the successor nodes currently connected to the task. The nodes in the tree cannot be modified directly. To add or remove successor nodes, use the layout to create and delete links. These changes are immediately reflected in the successor node tree.

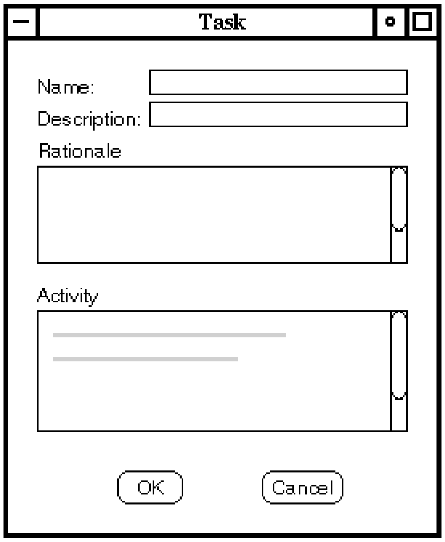


Figure L-4: Work Flow Manager Task Template

4.3 Activity Template

The activity node represents a single activity for a task and controls a single remote simulator. An activity only allows other activities as successor nodes. An activity can have either one task node or multiple activity nodes as predecessors. Multiple activity predecessor nodes allow creating an activity graph with interlocking dependencies. As each activity completes, its successor activities start. When all the activities complete, the encapsulating task node completes. Figure L-5 shows the activity template.

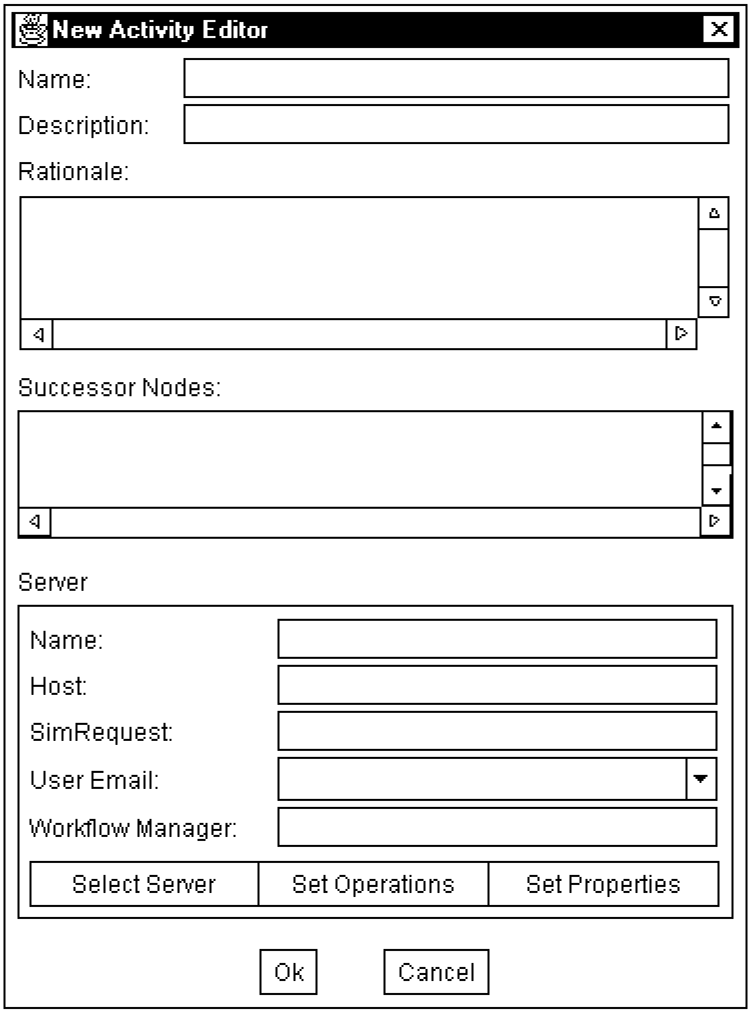


Figure L-5: Work Flow Manager Activity Template

The activity editor contains entries for the activity's name, a one-line description and a multiple line rationale. There is also a successor tree that expands to show all of the successor nodes currently connected to the activity. The nodes in the tree cannot be modified directly. To add or remove successor nodes, use the layout to create and delete links. These changes are immediately reflected in the successor node tree.

The activity editor also contains a remote server section. Within the server section is an area for the server's name, its host, the `SimRequest` property required by all remote simulators, and buttons to allow selecting a remote server (editor only), setting or viewing operations and setting or viewing properties. In order to edit the operations and properties, there must be a server name and host in the activity.

In order to accommodate activities that have requirements for user interaction, there are fields in the activity editor for user e-mail and a work flow process manager. The user e-mail field is a pull-down menu with e-mail addresses for selected tool users. The work flow process manager field is automatically populated with the e-mail address entered via the notify menu selection in the menubar. When a work flow progresses to an activity that has a user e-mail identified, the work flow automatically sends an e-mail message to that user with notification of the need to start that activity. At the same time, that work flow node is set to an internally paused state. Once the user is ready to proceed, he will resume the work flow and proceed with the simulation.

Selecting the server button brings up a list of all available remote servers known to the WFM. Selecting one from the list and pressing **Ok** makes that the remote server for the activity.
The operation button brings up an editor or viewer that shows all the operations and their commands. In the editor, you may select the command to use for an operation using the radio buttons. If no command is selected for an operation, the first command becomes the selection. The **Ok** button on the editor sets these operation/command pairs for the activity. Note that the operations are not sent to the remote server until run time so if there is a mistake in the commands, the error message appears at runtime and will have to be fixed then.

The property buttons brings up an editor or viewer that lists all of the remote server's properties and the values currently saved in the activity. All remote servers are required to have a `SimRequest` property. To modify a property, change its value and select the **Ok** button. Note that the properties are not sent to the remote server until run time so if there is a mistake in the data, the error message appears at runtime and will have to be fixed then.

5.0 Running a Work Flow

Once a work flow model is complete, it may be run and monitored from the layout area. As a work flow progresses, the colors of the nodes change on the layout. The colors indicate the node states and are defined as follows:

- **White** Reset state. The node is waiting to start.
- **Blue** Initialized and enabled state. The node is preparing for a launch.
- **Green** Working state. The node is working on its assignment.
- **Yellow** Pause state. The node is paused and waiting for a resume.
- **Gray** Completed state. The node is complete.
- **Red** Terminate state. The node is stopped and cannot resume without a reset.
- **Orange with Red Outline** Faulty state. The node terminated and cannot resume without a reset. One cause of this is issuing an invalid operation or command to the server.

To run a model, choose the appropriate operation from the menubar or toolbar and select a node for the operation. A run operation may be applied to any node in a model, thus allowing flexibility such as pausing one branch of a model while another one continues to operate. There are five operations provided as part of the Work Flow Manager – start, pause, resume, terminate, and reset.

Start

To start a work flow model, select the start operation from the menubar or toolbar and click on the node to start. This node becomes the "initial" node, which means it is the top node of the run and all operations take place below it. An initial start node is indicated by its name displaying in boldface text. The WFM supports multiple start nodes so portions of a large process model may be run independently of each other. To start executing an entire model, start the top-most node in the model.

Pause

To pause a work flow model, select the pause operation from the menubar or toolbar and click on a node. All predecessor nodes directly in line of the paused node and all successor nodes go into the pause state. The predecessor nodes do not send the pause command down to their successor nodes so any parallel branches above the paused node continue execution. All successor nodes below the paused node get the pause command and enter the pause state. If a remote simulator does not support the pause command, its associated activity generates an error message and the branch continues to execute.

The pause state set by the work flow for sending an e-mail message to the user of an interactive tool, unlike the pause state described above, does not communicate with the remote simulator. This state is internal to the work flow manager and must be resumed by either the tool user or the process manager before any further messages are sent to the simulator.

Resume

To resume a work flow model, select the resume operation from the menubar or toolbar and click on a node. All successor nodes receive a resume command and continue their execution. The resume command is not sent up to the predecessor nodes so any paused predecessors stay in the paused state.

Terminate

To terminate a work flow model, select the terminate operation from the menubar or toolbar and click on a node. All predecessor nodes directly in line of the terminated node and all successor nodes go into the terminate state. The predecessor nodes do not send the terminate command down to their successor nodes so any parallel branches above the terminated node continue execution. Once terminated, a node can only be reset and restarted again. Terminate should only be used to halt a work flow model that has gone awry.

Reset

To reset a work flow model, select the reset operation from the menubar or toolbar and click on a node. That node and all of its successor nodes are reset. There is no "reset" operation on a remote simulator; resetting only applies to the process model. Since a remote simulator cannot reset, it may continue execution and its state changes show up on its activity. Once reset, a node will accept a start operation again. To reset an entire process model, reset the top-most node.

6.0 Saving and Loading Work Flow Models

Saving and loading a work flow model is simple. Select the save menubar item or toolbar icon and the current work flow model is saved with the same name. If there is no name yet (it is a new model), a message appears in the status bar asking the user to use the "Save as..." operation. Selecting "Save as..." displays a file dialog box where the user selects a file or enter a new one. Selecting **Ok** saves the work flow model while **Cancel** aborts the save.

To load a work flow model, select the load menubar item or toolbar icon and a dialog box appears asking whether to erase the current work flow. If **Yes** is selected, the current model is deleted before the load. If **No** is selected, the new model is added to the existing model, which allows for building a large model from smaller ones.

After the selection is made regarding the existing model, a file dialog box appears. Selecting **Cancel** aborts the operation and leaves the current model intact. Selecting a work flow model file to load and pressing **Ok** loads the model into the WFM.

The saved work flow models will develop dependencies among themselves. This is an artifact of using serialization. If a model is moved or deleted that others are dependent on, then the dependent models will not load properly. The best way to avoid this potential problem is to keep models together once they are created, delete only groups of models, modify instead of delete where possible, and copy instead of move models that are to be used in other work flows.

7.0 Sample SAVE Work Flow

For the SAVE Interim Demonstration, the Work Flow Manager was used to develop a process flow for the simulations. Information about the activities that were planned, the dependencies of the activities, and pertinent information about which information to use at each step of the process were recorded. The work flow for this demonstration is shown in Figure L-6. The team decided to use the gunport alternative as the highest level process and include all of the activities involved in evaluating that alternative in a single work flow. For larger studies, the work flow may be divided into individual models and merged together at the appropriate time. There were two primary tasks in the process. The first created the initial process planning information. The second task involved refining the plan and updating information about cost and schedule. The individual activities are labeled according to the tool they use and the function they are performing. Naming conventions for the nodes are left to user discretion; however, it is helpful to give meaningful names to each node.

This particular work flow was started at the task level. This allowed user control over the execution of the activities below the second task. The team selected this method of work flow control in order to accommodate two tools that were not yet integrated with the WFM. These tools performed their simulations and communicated with the SAVE data model outside the control of the WFM. Once their data was successfully transferred, the second node of the work flow was started.

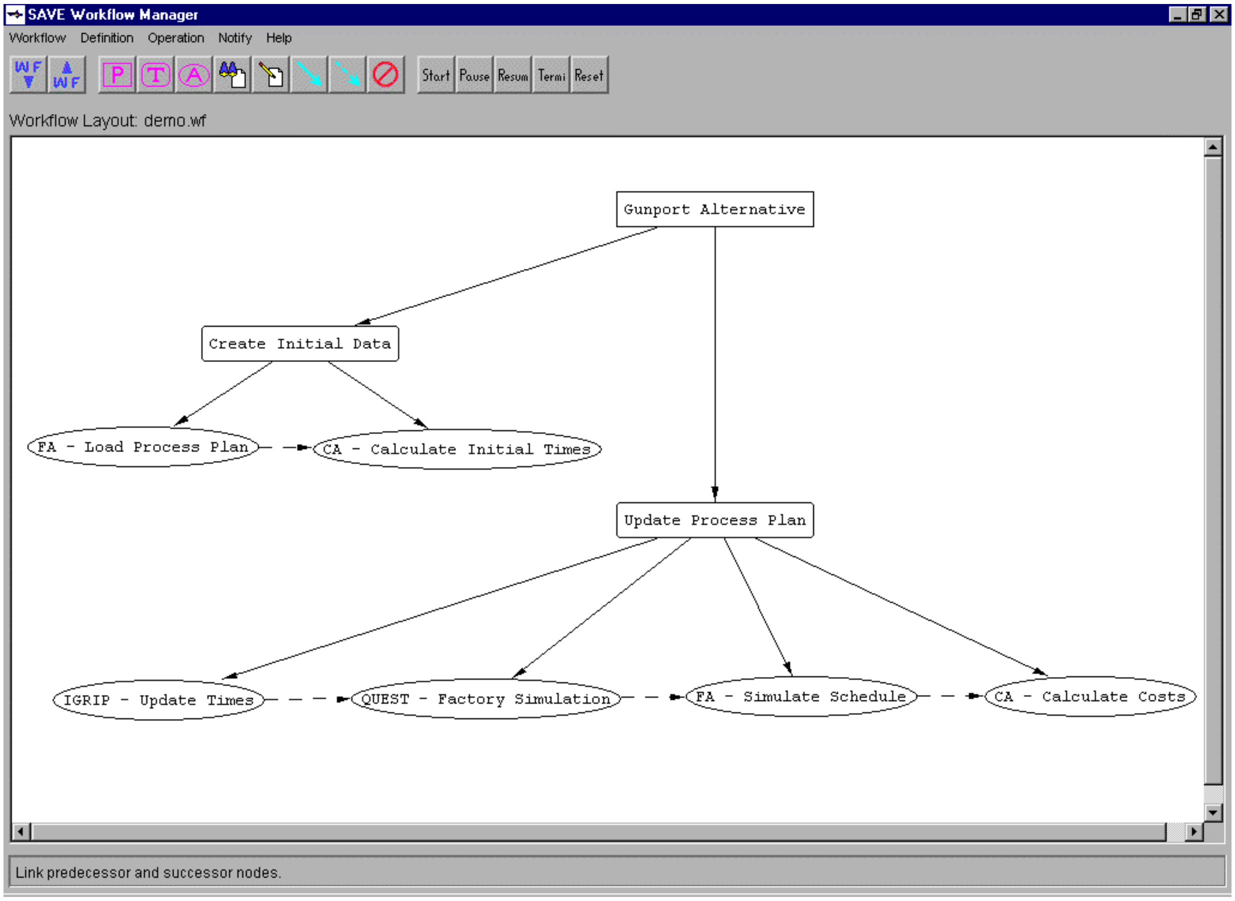


Figure L-6: Sample Work Flow from SAVE Interim Demonstration

8.0 Interactions

The objects within the WFM communicate with the various simulators. This section details the major interactions between the WFM internal objects and the external simulators. It is not necessary for typical Work Flow Manager users to understand these interactions in detail; however, this information may provide useful insight into how the Work Flow Manager operates. The interactions are presented in diagrams that show the flow of information and the corresponding actions among the components of the system. Even though only one process and task are shown in the diagrams, there can be many processes and tasks in a work flow. Reference is made to the other objects where appropriate.

In the diagrams, text between quote marks indicate a state change while text followed by parentheses represent an action. The curved lines within an object's activation period means there is some delay (possibly manual) during that interval.

The **Starting a Process** diagram in Figure L-7 shows how a work flow proceeds from start to end. It is always initiated from the WFM. In this case, there are no pauses, terminations or faults in the process. The other diagrams show the handling of these types of actions.

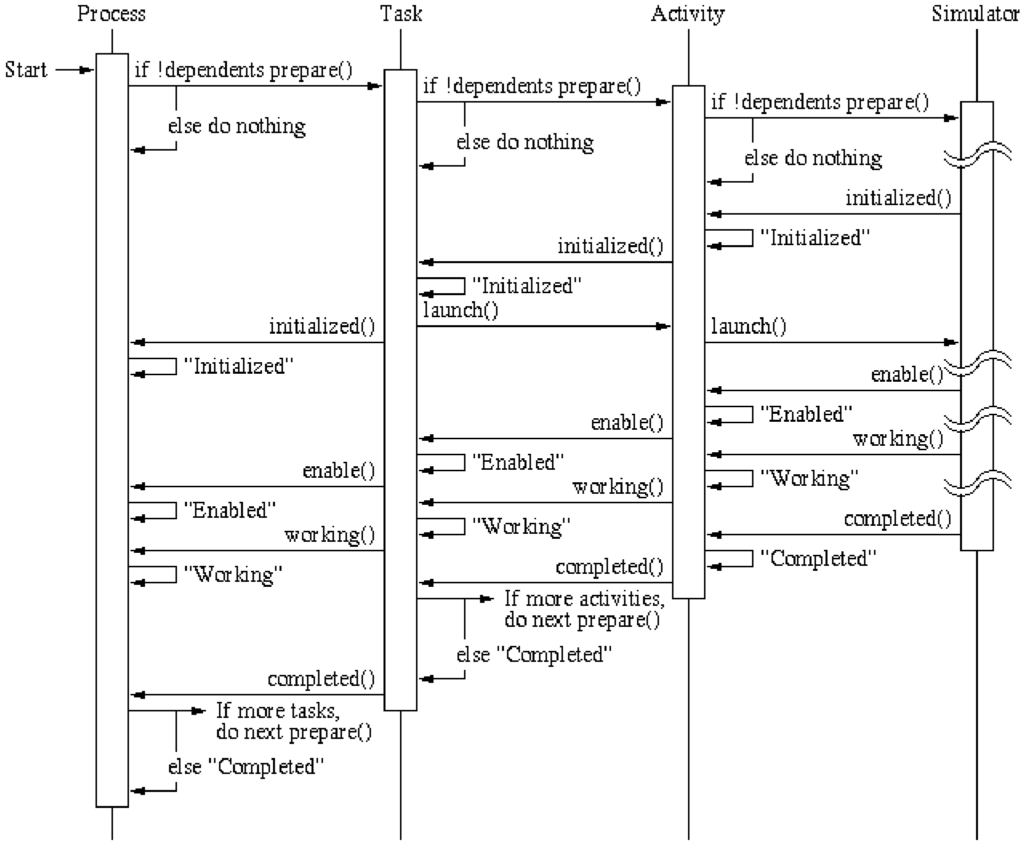


Figure L-7: Interaction Diagram for Starting a Process

The **Pausing a Working Process** diagram in Figure L-8 illustrates the steps in pausing and resuming a process node from the WFM. If a simulator initiates the action, then start from the simulator node side.

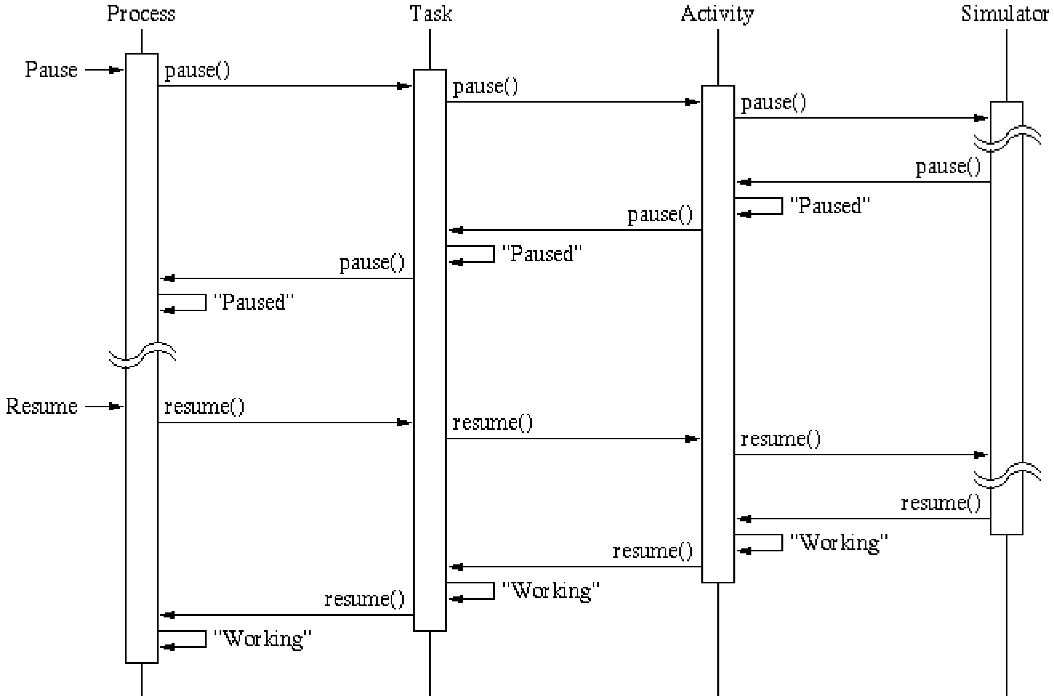


Figure L-8: Interaction Diagram for Pausing a Working Process

The **Terminating a Working Process** diagram in Figure L-9 shows the steps in terminating a process node from the WFM. If a simulator initiates the action, then start from the simulator node side. Note that a terminate does not stop any processes that are running in parallel.

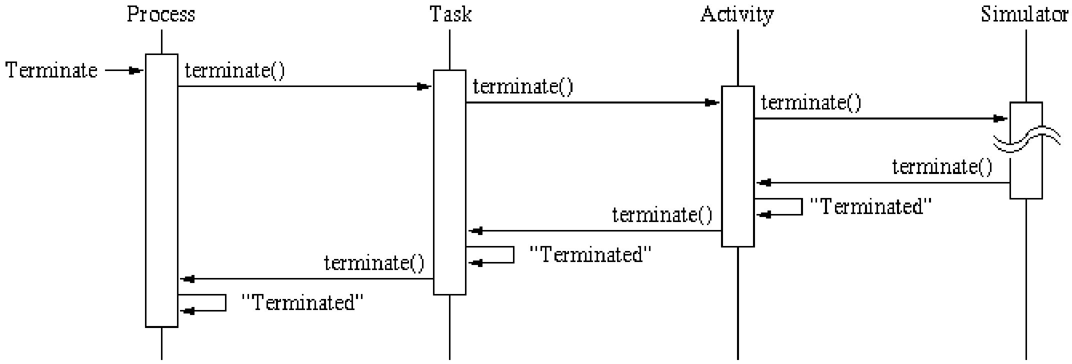


Figure L-9: Interaction Diagram for Terminating a Working Process

The **Handling a Simulator Failure** diagram in Figure L-10 shows what happens when a simulator generates a fault. The WFM never generates a fault event; only a simulator is capable of that. Note that a fault does not terminate any processes that are running in parallel.

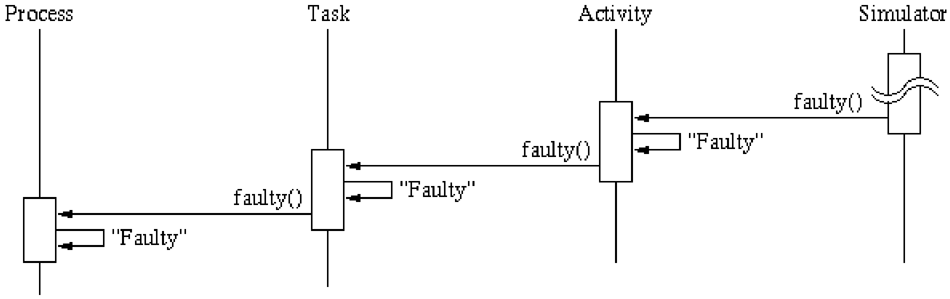


Figure L-10: Interaction Diagram for Handling a Failure

Appendix M

Query Manager User's Guide

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

1.0 Introduction

When employing an integrated toolsuite, it is important for users to have the ability to access the information that is being shared among the tools. The SAVE infrastructure contains a Query Manager (QM) application to provide visibility into that information in the SAVE data model (SDM) without accessing one of the simulation tools.

The QM is a Java/CORBA application for browsing, creating, modifying and deleting objects in the SDM. This application does not interface directly with either the simulation tools or the WFM. Its sole purpose is to provide access to information in the SDM through a mechanism other than the simulation tools within the SAVE toolsuite. The QM utilizes the library objects within the SDM and displays them in a tree-like structure.

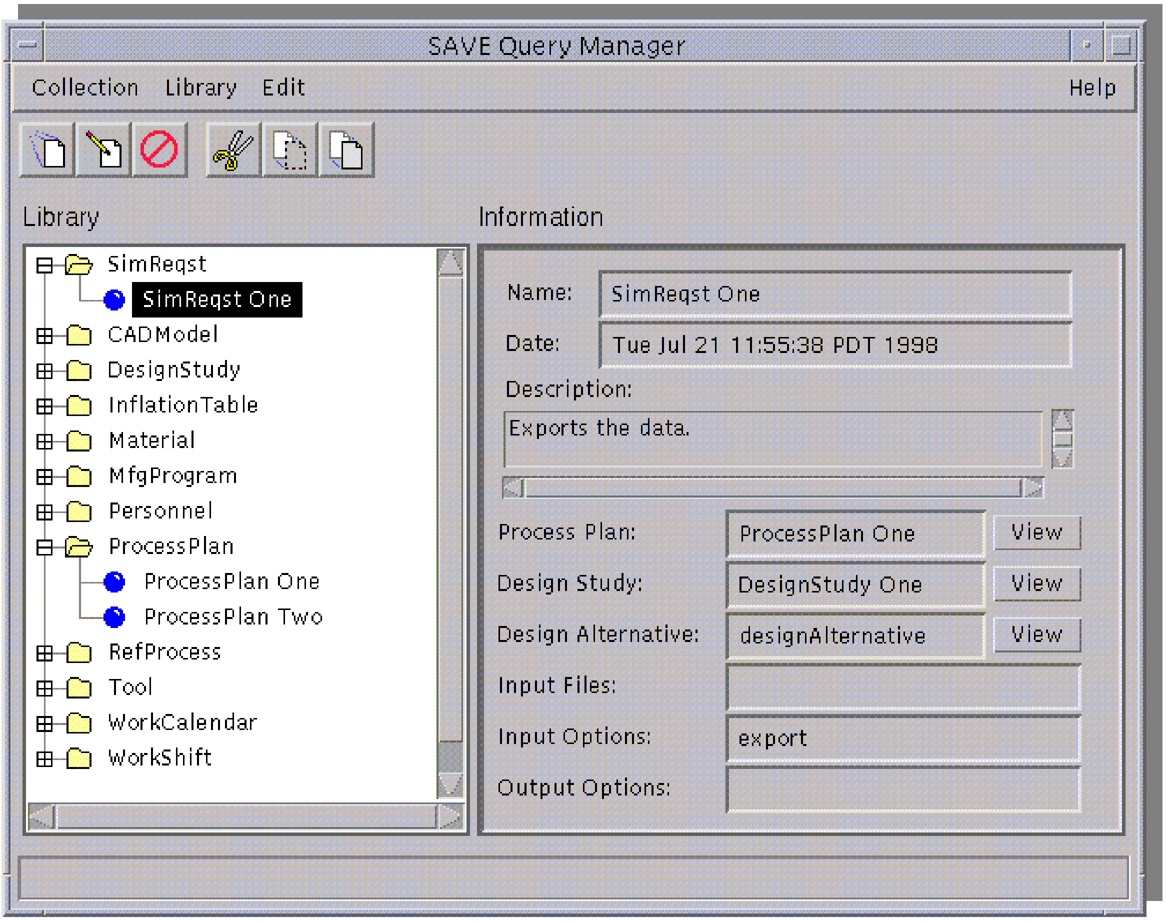
The QM uses a point-and-click interface to access information in the SDM. SAVE libraries are selected within the tree window, while actions are highlighted from a menubar or toolbar that specify whether to create, edit or modify an object in the library. Separate popup windows are displayed depending on which function is selected. When modifications are made to an object, the Query Manager uses the transaction management provided by the SDM to insure consistency in the data.

2.0 Starting the Query Manager

The QM takes only one argument. To start the QM, double click on the QM icon installed on your computer. Upon installation, this icon should have been configured with the server argument that specified the IP address and name for the SAVE server.

3.0 Query Manager Interface

The QM interface consists of a menubar, toolbar, library window, display window and status bar. Figure M-1 shows this interface.



Name: SimReqst One

Date: Tue Jul 21 11:55:38 PDT 1998

Description: Exports the data.

Process Plan: ProcessPlan One

Design Study: DesignStudy One

Design Alternative: designAlternative

Input Files:

Input Options: export

Output Options:

Figure M-1: Query Manager User Interface

The menubar is along the top of the window with a toolbar that implements the primary user tasks just below it. The library window, which provides a list of the available library objects, is in the left center of the interface. To the right of the library window is the information window. This window shows the current viewing selection from the library. Under the library and information windows is the status bar, which displays both the toolbar tips and any messages and usage prompts.

3.1 Menubar

The menubar controls the QM and consists of four primary selections: collection, library, edit and help. These selections are shown in Figure M-2. To select a menubar item, click on the top-level selection and then the desired item. To cancel the operation, just move the cursor off the menubar and click. Selections that are not yet implemented are identified as inactive in the descriptions that follow.

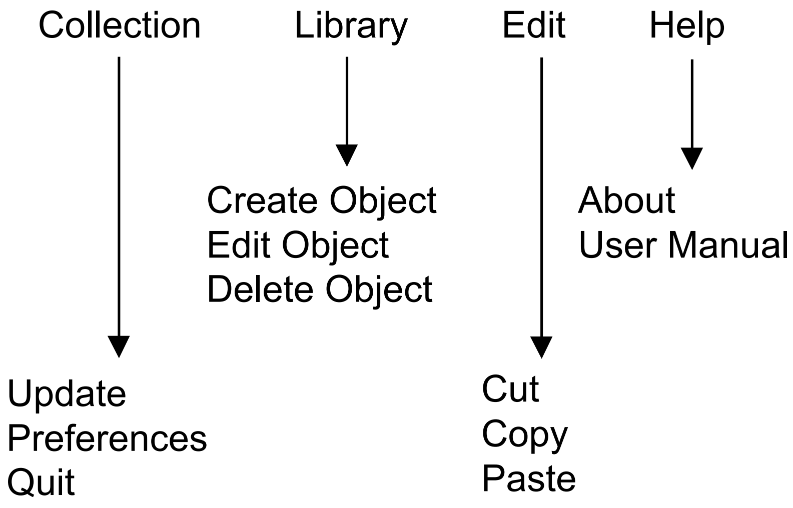


Figure M-2: Query Manager Menubar

3.1.1 Collection

The collection menu item contains selections that allow the user to update information or set preferences in the QM application. The selections and their definitions are as follows:

- **Update** *(inactive)* Update the information in the current QM windows.
- **Preferences** *(inactive)* Set any user preferences.
- **Quit** Quit the application.

3.1.2 Library

The library menu item contains selections that allow the user to modify information in the SAVE database. The selections and their definitions are as follows:

- **Create Object** Create a new library object.
- **Edit Object** Edit an existing library object.
- **Delete Object** Delete an existing library object.

3.1.3 Edit

The edit menu item contains selections that allow the user to copy and paste text information from one field to another. The selections and their definitions are as follows:

- **Cut** Cut (remove) the selected text.
- **Copy** Copy the selected text.
- **Paste** Paste text that has be cut or copied into the selected field.

3.1.4 Help

The help menu item contains selections that allow the user to obtain information about the use of the query manager application. The selections and their definitions are as follows:

- **About** Display the about Query Manager message.
- **User manual** *(inactive)* Display the user manual.

3.2 Toolbar

The toolbar contains icons that represent the major operations on the menubar. They are separated into three groups that correspond to three top-level menubar items. The groups and icons from left to right are:

- **Library**
    - **Create Object**
    - **Edit Object**
    - **Delete Object**
- **Edit**
    - **Cut**
    - **Copy**
    - **Paste**

Passing the cursor over a toolbar icon displays its tool tip in the status bar, which provides information about its identity. Moving the cursor off the toolbar icon restores the message that was previously in the status bar.

The way a toolbar icon acts depends upon its function. For those that perform their operation in one step, the icon works like a button. Most of the toolbar icons, however, perform multiple step operations. These toolbar icons "freeze" in the down position and the next step in the sequence appears in the status bar. To cancel one of these multiple step operations, click the toolbar icon a second time.

3.3 Library Window

The library window provides a tree structure display of the libraries in the SDM. The names of the libraries are listed as folders in the hierarchy. The following libraries are available via the QM:

- SimReqst
- Break
- CADModel
- DesignStudy
- InflationTable
- Material
- MfgProgram
- Part
- Personnel
- ProcessPlan
- RefProcess

- Tool
- WorkCalendar
- WorkShift

Traversing the library tree is similar to using directory listing in the Windows operating system. Each folder, or library, has a + sign to the left of it. Clicking on the + sign will expand the tree and display each instance of that particular library object that exists in the SAVE database. When the tree is expanded, the + sign changes to a – sign. The object instances are preceded by a blue sphere and are identified by their name attribute. The +/– signs work like a toggle switch. To collapse the list, click on the – sign.

3.4 Information Window

The information window provides detailed information about a specific object in the SAVE database. It contains fields for each of the attributes of a specific object. In cases where an attribute is another object, there is a "view" button to the right of the attribute. Clicking on this button displays that object in a separate popup information window. In cases where an attribute is a sequence or list of other objects, there is a "list" button to the right of the "view" button. Clicking on this button displays a pick list of the objects in the sequence. Any object in the list may be selected with a left mouse click and displayed with the "view" button.

To display information about a particular object in the information window, select that object in the library window. When another object is selected from the library window, the information window refreshes with the attributes of that object.

3.5 Status

The status bar is at the bottom of the display and shows tool tips, messages and usage prompts. A tool tip shows when the cursor passes over a toolbar icon. The messages pop up as determined by the system. Error messages always beep when they display to draw attention to themselves. Usage prompts appear during the multiple step operations such as creating a new library object.

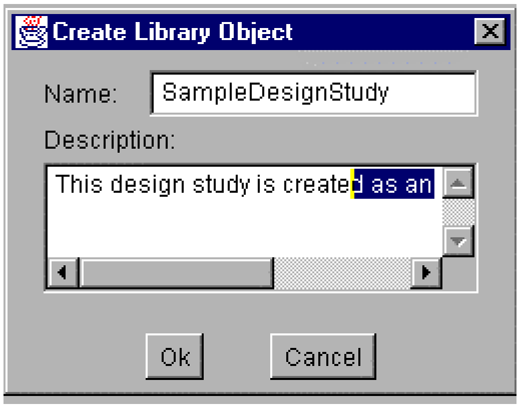
4.0 Creating and Modifying Data

There are three options available in the Query Manager interface for interacting with the SAVE data – create, edit, and delete. To work with any of these options, select the appropriate action from either the menubar or the toolbar. A prompt will appear in the status window telling the user to select either a library or object within the library. For object creation, select the library of the desired object type. For object editing or deletion, select the specific object within the library.

The Query Manager uses the transaction management scheme that is built into the SAVE database server. Changes are "committed" to the database as they are made within the Query Manager. Essentially, clicking the OK button on any create, modify or delete screen will immediately make those changes in the database.

4.1 Create Object

Once the create object selection is made and the appropriate library is highlighted, a create library object popup window, shown in Figure M-3, is displayed. The window allows the user to enter the name and description of the library object. Other attributes are not available through the create library object window. The values for these attributes must be populated using the edit object function once the object is created.



Create Library Object

Name: SampleDesignStudy

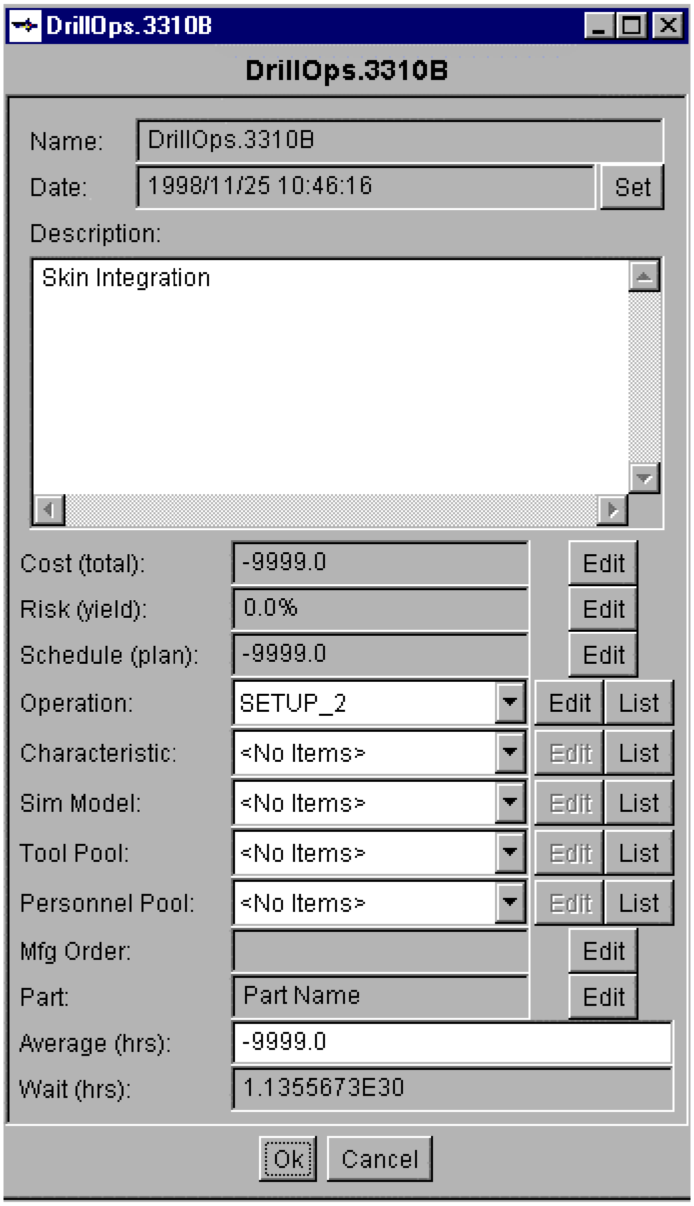
Description: This design study is created as an

Figure M-3: Query Manager Create Template

4.2 Edit Object

Once the edit object selection is made and the appropriate object is highlighted, an object editing window is displayed. The edit window layout varies for each type of object. It contains fields that represent each attribute of that specific object. If a value is present in the database, it is displayed in the appropriate field. An empty field signifies that there is no value for that attribute in the database. For newly created objects, only the name and description fields are present. Figure M-4 shows the edit window for a specific process plan object with the name DrillOps.3310B.

Several types of buttons are available on the edit window. There is a set button that allows the user to reset the date/time stamp for that object. Typically, the date/time is set automatically at the time the object is created. For attributes that are other objects, there is an edit button that allows the user to modify that particular object without going back to the library. In addition, this option is available for objects that are not available through the library. Figure M-5 shows the result of editing the Setup_2 operation in the DrillOps.3310B process plan. For attributes that are a sequence, or list, of other objects, there are two buttons – edit and list. The edit button allows the user to modify the object that is currently shown in the field. The list button displays a list of the objects in the sequence. To select a different object, use the mouse to highlight the desired object. Figure M-6 shows the result of displaying a list of the operations for the process plan. The list window also provides a mechanism for adding and removing items from the list. To remove an item, select the item from the list and click the remove button. Clicking the add button will display a popup window that creates a new item that will be added to the list. Once all modifications are complete, click the OK button to register the changes. Clicking cancel will cause the object list to revert back to its original state.



DrillOps.3310B

Name: DrillOps.3310B

Date: 1998/11/25 10:46:16

Description:

Skin Integration

Cost (total): -9999.0

Risk (yield): 0.0%

Schedule (plan): -9999.0

Operation: SETUP_2

Characteristic: <No Items>

Sim Model: <No Items>

Tool Pool: <No Items>

Personnel Pool: <No Items>

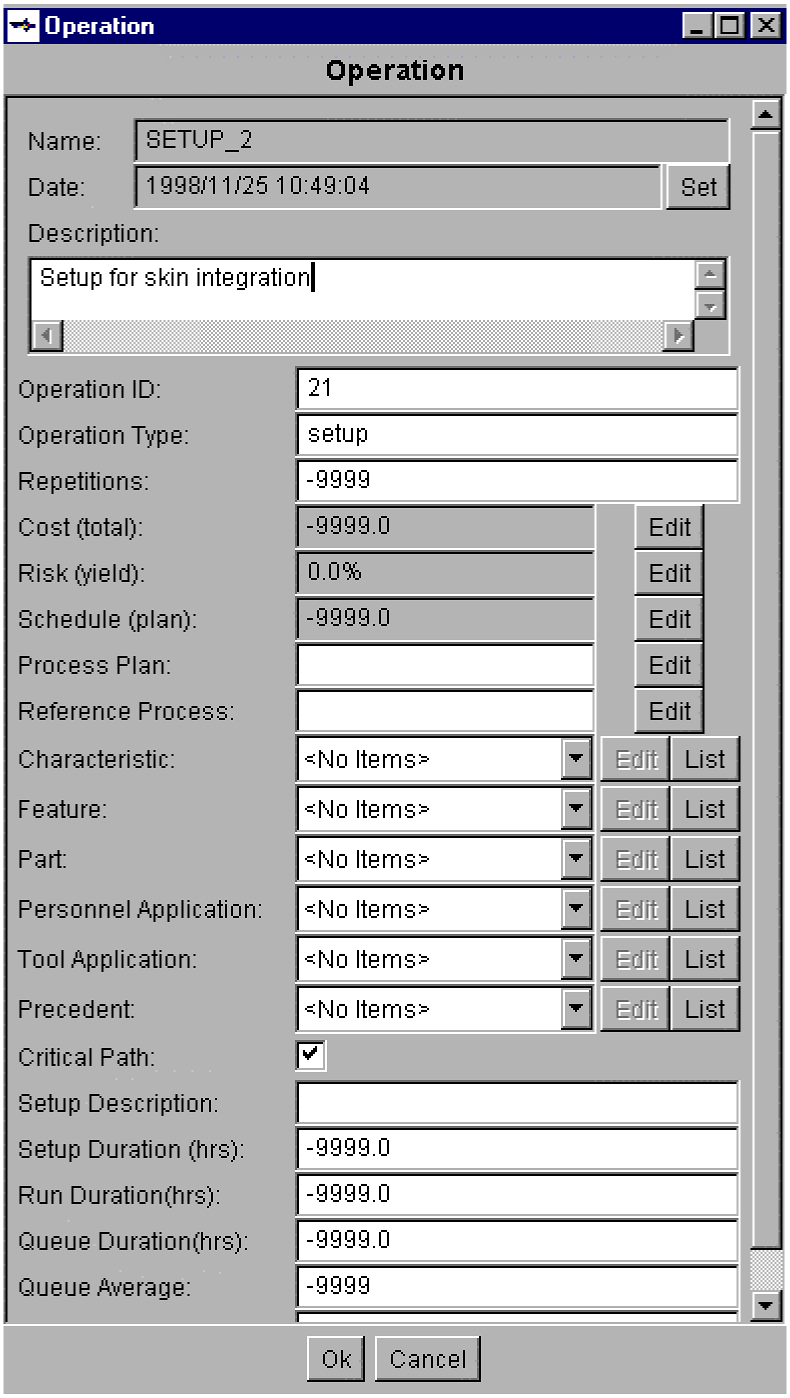
Mfg Order:

Part: Part Name

Average (hrs): -9999.0

Wait (hrs): 1.1355673E30

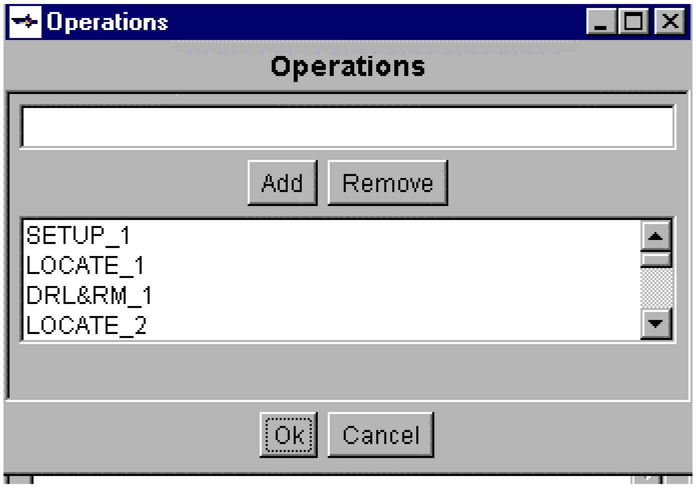
Figure M-4: Query Manager Edit Template



Operation

Name:	SETUP_2
Date:	1998/11/25 10:49:04
Description:	Setup for skin integration
Operation ID:	21
Operation Type:	setup
Repetitions:	-9999
Cost (total):	-9999.0
Risk (yield):	0.0%
Schedule (plan):	-9999.0
Process Plan:	
Reference Process:	
Characteristic:	<No Items>
Feature:	<No Items>
Part:	<No Items>
Personnel Application:	<No Items>
Tool Application:	<No Items>
Precedent:	<No Items>
Critical Path:	☑
Setup Description:	
Setup Duration (hrs):	-9999.0
Run Duration(hrs):	-9999.0
Queue Duration(hrs):	-9999.0
Queue Average:	-9999

Figure M-5: Result of Edit Button



SETUP_1
LOCATE_1
DRL&RM_1
LOCATE_2

Figure M-6: Result of List Button

4.3 Delete Object

Once the delete object selection is made and the appropriate object is highlighted, message window is displayed to confirm that this object is to be deleted. Select OK to continue with the delete, or select cancel to leave the object intact. The user should take care in the use of the delete object option, because there is not an "undo" capability within the Query Manager application.

Even though the user deletes an object in the Query Manager, the object may not be completely deleted from the SAVE database. The SAVE server manages object usage internally and keeps track of where library objects are used. If an object is still being used elsewhere, the object is not totally deleted from the database, only its reference is removed from the library. If the user wants to insure that the object is deleted, he must first delete it wherever it is used and then delete it from the library.

The user should exercise caution when using the delete function, especially in lists that allow duplicate names, since there can be confusion over two separate objects that have identical names.

Appendix N

Training Material

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

Simulation Assessment Validation Environment

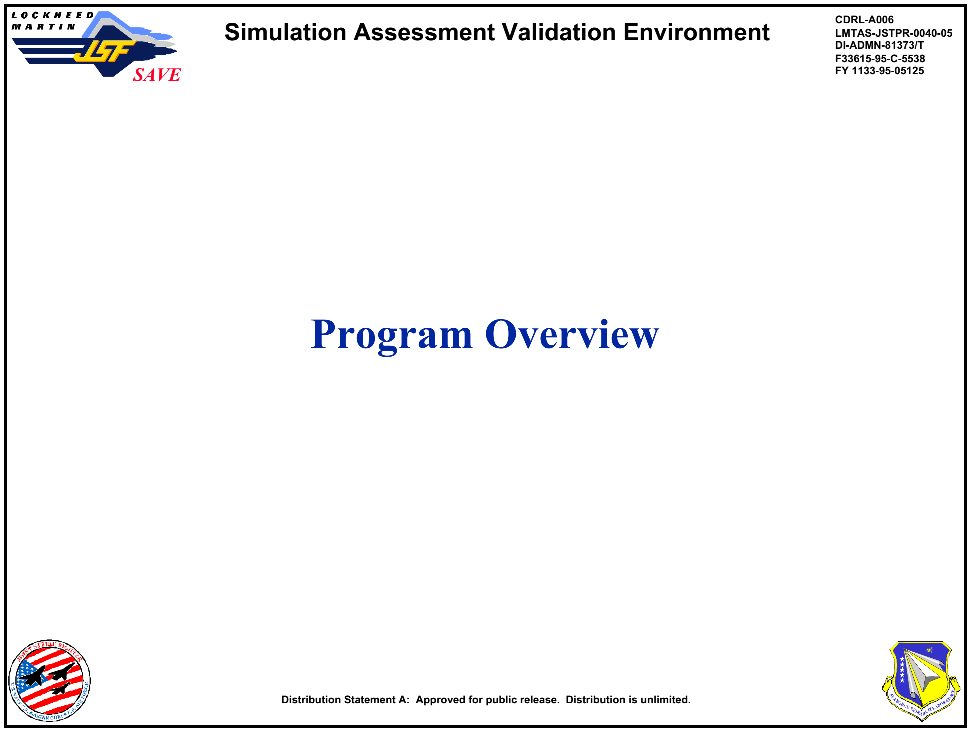
CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

SAVE

Training

SAVE
DESIGN
SIMULATE
MANUFACTURE
JSF • Affordable by Design

Distribution Statement A: Approved for public release. Distribution is unlimited.



Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Program Overview

Distribution Statement A: Approved for public release. Distribution is unlimited.

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

The Vision

Making Virtual Manufacturing A Reality

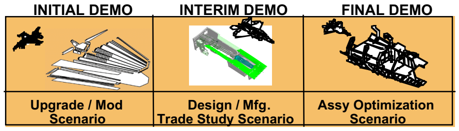
- Virtual manufacturing simulations will improve product affordability by :
  - Identifying better product/process alternatives
  - Solving potential manufacturing problems early in design
  - Reducing scrap, rework, repair, and redesign
- Integration of manufacturing simulation tools is necessary to make their use timely and affordable

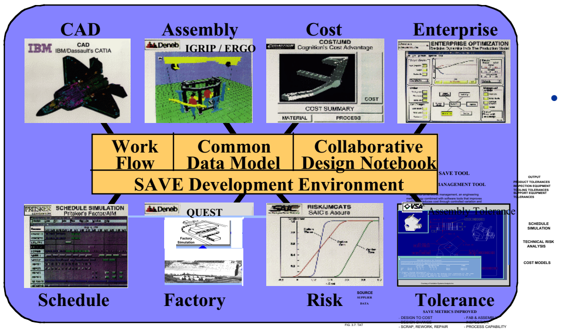
Distribution Statement A: Approved for public release. Distribution is unlimited.

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

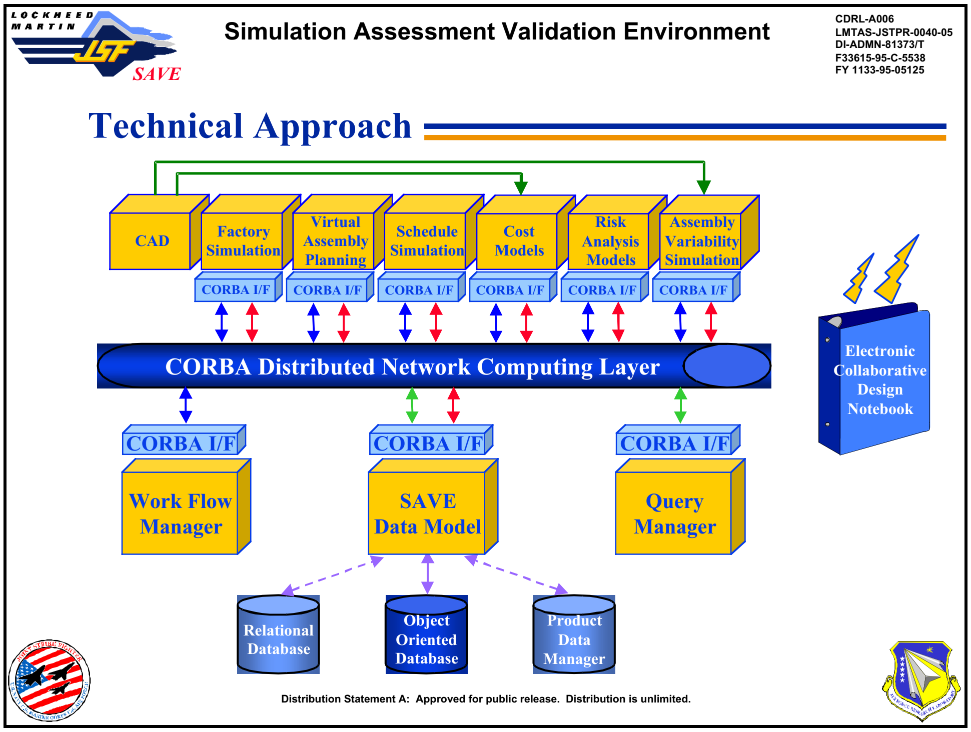
The Reality


INITIAL DEMO
INTERIM DEMO
FINAL DEMO
Upgrade / Mod Scenario
Design / Mfg. Trade Study Scenario
Assy Optimization Scenario



- Three Demonstrations
  - Develop concept of operations
  - Validate savings metrics
  - Test the integration infrastructure
- Infrastructure
  - CORBA-based Manufacturing Simulation Data Model
  - Workflow Manager
  - Collaborative Electronic Design Notebook
  - Data Model Editor

Distribution Statement A: Approved for public release. Distribution is unlimited.



LOCKHEED MARTIN JSF SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Technical Approach

CAD

Factory Simulation

Virtual Assembly Planning

Schedule Simulation

Cost Models

Risk Analysis Models

Assembly Variability Simulation

CORBA I/F

CORBA I/F

CORBA I/F

CORBA I/F

CORBA I/F

CORBA I/F

CORBA Distributed Network Computing Layer

CORBA I/F

Work Flow Manager

CORBA I/F

SAVE Data Model

CORBA I/F

Query Manager

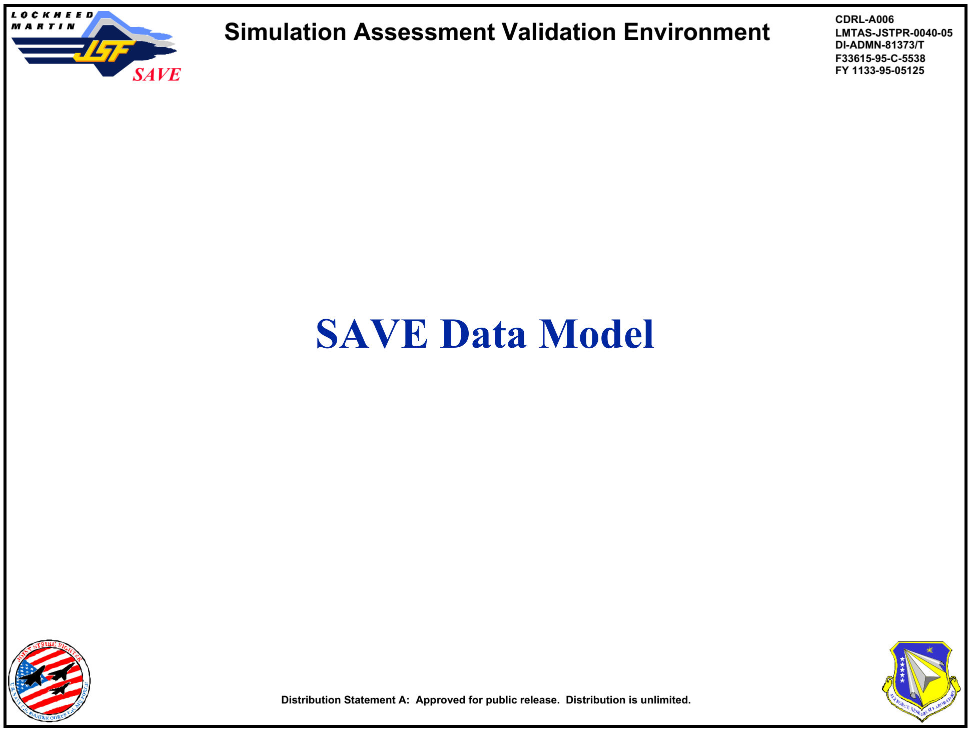
Relational Database

Object Oriented Database

Product Data Manager

Electronic Collaborative Design Notebook

Distribution Statement A: Approved for public release. Distribution is unlimited.



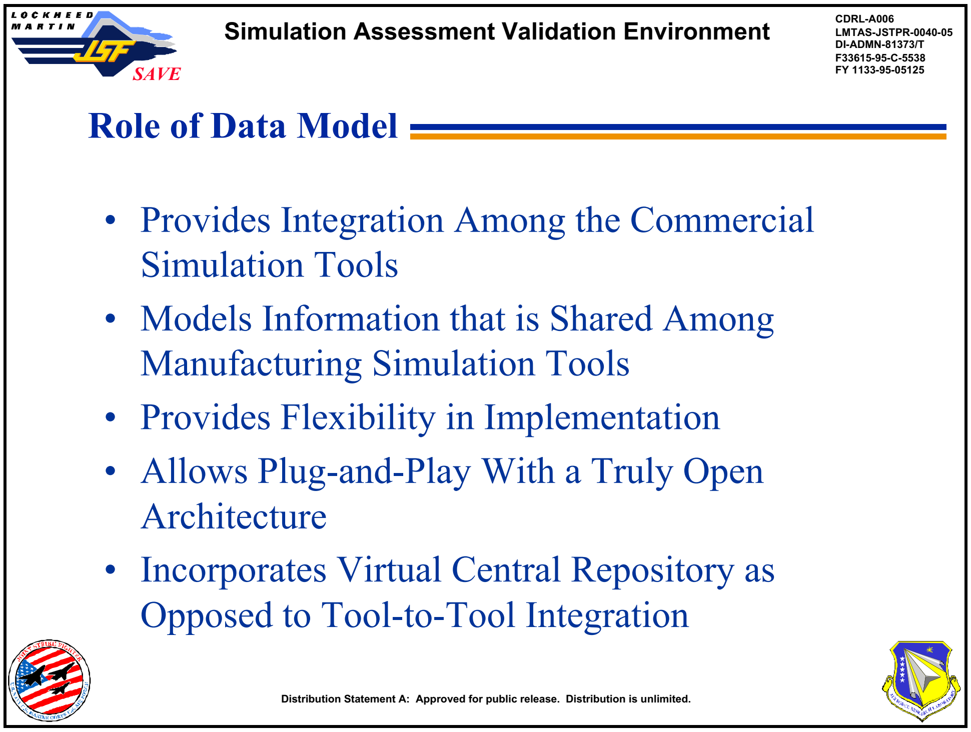
LOCKHEED MARTIN JSF SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

SAVE Data Model

Distribution Statement A: Approved for public release. Distribution is unlimited.



SAVE

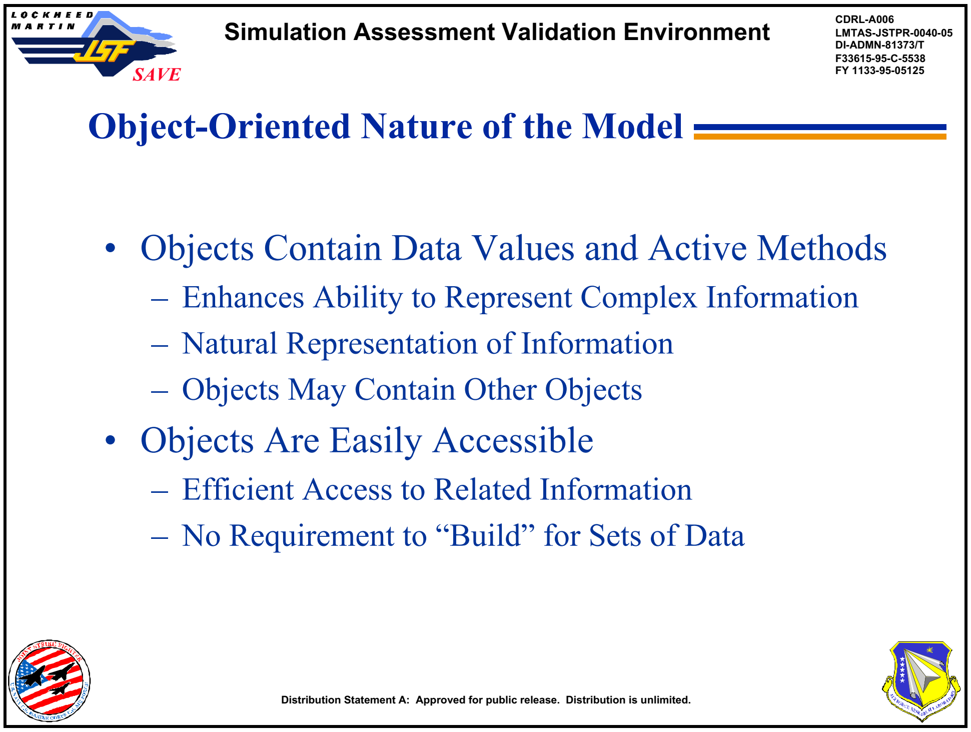
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Role of Data Model

- Provides Integration Among the Commercial Simulation Tools
- Models Information that is Shared Among Manufacturing Simulation Tools
- Provides Flexibility in Implementation
- Allows Plug-and-Play With a Truly Open Architecture
- Incorporates Virtual Central Repository as Opposed to Tool-to-Tool Integration

Distribution Statement A: Approved for public release. Distribution is unlimited.



SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Object-Oriented Nature of the Model

- Objects Contain Data Values and Active Methods
  - Enhances Ability to Represent Complex Information
  - Natural Representation of Information
  - Objects May Contain Other Objects
- Objects Are Easily Accessible
  - Efficient Access to Related Information
  - No Requirement to “Build” for Sets of Data

Distribution Statement A: Approved for public release. Distribution is unlimited.

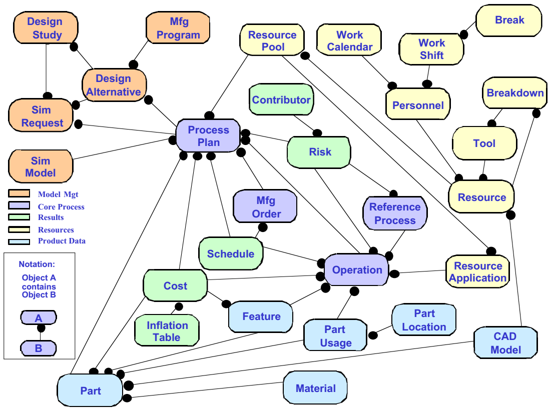
LOCKHEED MARTIN JSF SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

SAVE Data Model

- Inputs from Numerous Sources
  - Manufacturing Engineers
  - Design Engineers
  - Simulation Software Vendors
  - Simulation Software Users
  - Previous Studies
- Includes Six Types of Data
  - Core Process
  - Resource
  - Part
  - Results
  - Model Management
  - Utility (not shown)



Distribution Statement A: Approved for public release. Distribution is unlimited.

LOCKHEED MARTIN JSF SAVE

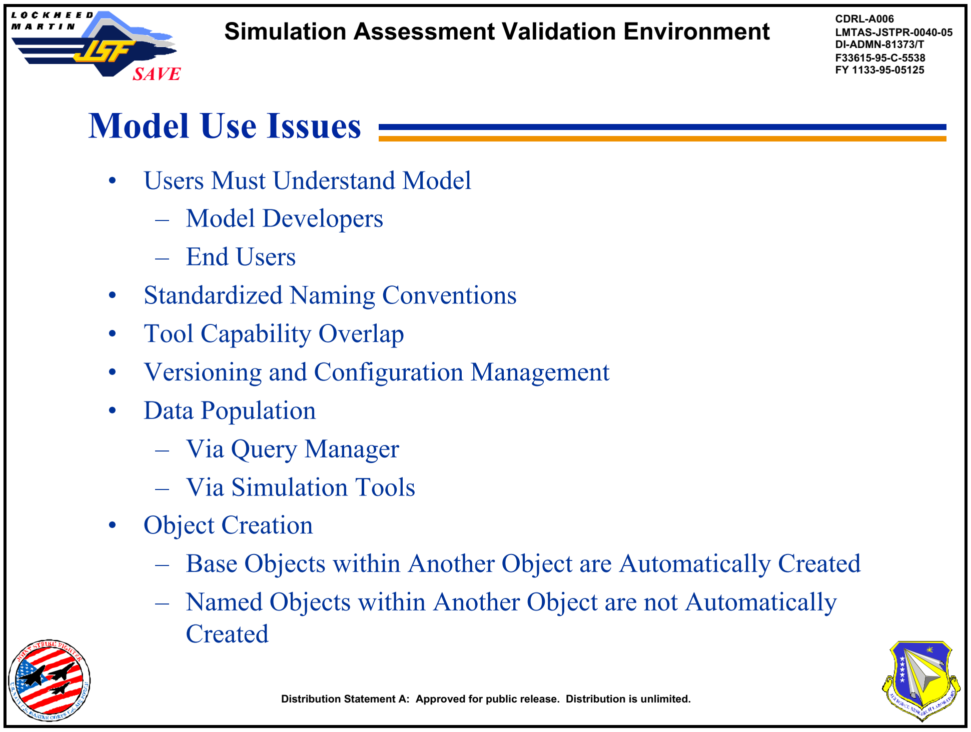
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Data Model Web Pages

- Use for Training Purposes
- Hyperlinked Web Pages that Operate Like the Object Oriented Model Itself
- Contents of Pages
  - Textual Description of Object
  - List of Data Fields, Their Type, and Description
  - List of Methods Supported and Their Function
  - Hyperlinks Where Appropriate

Distribution Statement A: Approved for public release. Distribution is unlimited.



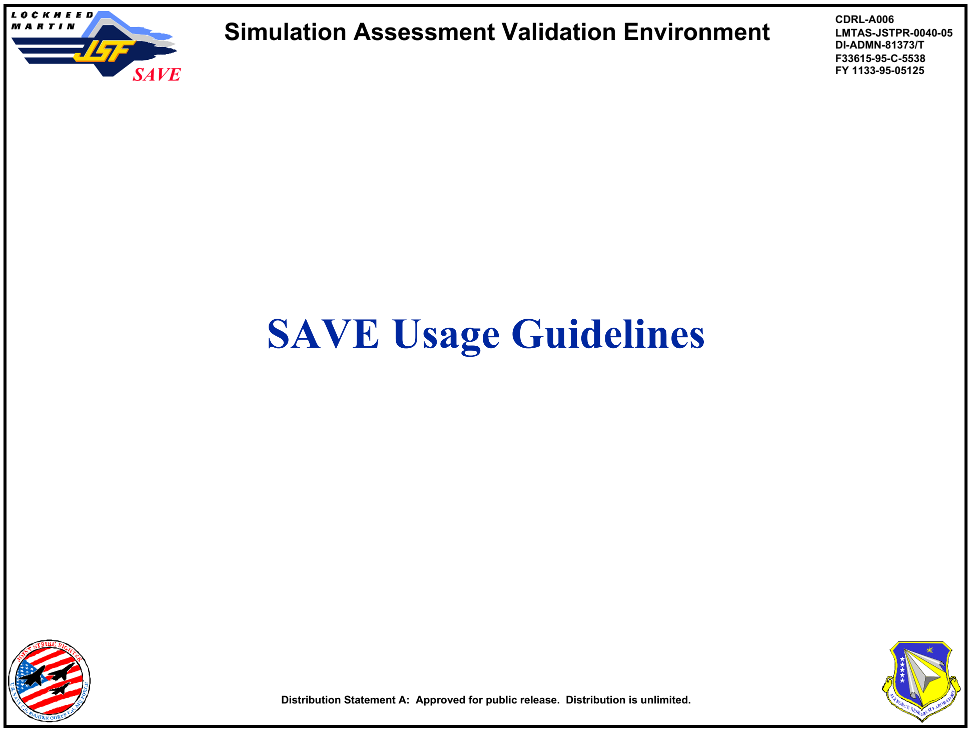
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Model Use Issues

- Users Must Understand Model
  - Model Developers
  - End Users
- Standardized Naming Conventions
- Tool Capability Overlap
- Versioning and Configuration Management
- Data Population
  - Via Query Manager
  - Via Simulation Tools
- Object Creation
  - Base Objects within Another Object are Automatically Created
  - Named Objects within Another Object are not Automatically Created

Distribution Statement A: Approved for public release. Distribution is unlimited.



Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

SAVE Usage Guidelines

Distribution Statement A: Approved for public release. Distribution is unlimited.

SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Inderdisciplinary Interactions

SAVE Integrated Virtual Manufacturing Environment Requires Interactions and Coordination Among Many Disciplines

- Project Engineering
- Manufacturing
- Systems Engineering
- Industrial Engineering
- Design Engineering
- Business Operations

SAVE TOOL USAGE	TOOL							
DISCIPLINE	CAD	FACTORY SIM..	ASSEMBLY CELL SIM.	ERGONOMICS	SCHEDULE SIM.	COST MODELING	RISK MODELING	DIMENSIONAL MGT
STRUCTURAL DESIGN	●		●	●		●		●
MECHANICAL DESIGN	●		●	●		●		●
ELECTRICAL ENGINEERING	●		●	●				
TOOL DESIGN	●		●	●				●
NC PROGRAMMING	●							
FACILITIES ENGINEERING	●	●	●					
PRODUCIBILITY ENGINEERING	●		●	●	●	●	●	●
QUALITY ENGINEERING	●		●	●	●	●	●	●
SUPPORTABILITY ANALYSIS	●		●		●			
INDUSTRIAL ENGINEERING		●	●	●		●	●	●
PLANNING		●	●	●				●
MASTER SCHEDULING					●			
FINANCE						●		
VALUE ENGINEERING						●		
MANUFACTURING ENGINEERING	●	●	●	●				●
IPPT LEAD		●	●		●	●	●	
PROGRAM MANAGER		●			●	●	●	

TABLE 3.1: Engineering and manufacturing discipline utilization of the SAVE system.

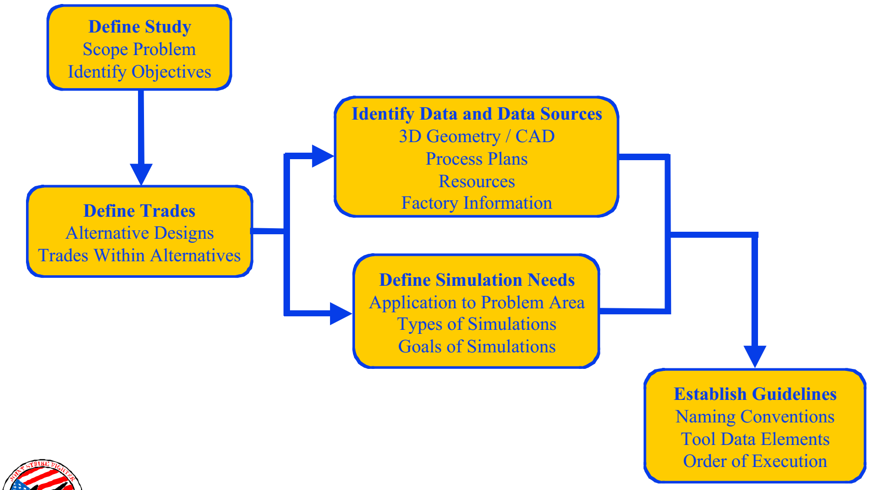
Distribution Statement A: Approved for public release. Distribution is unlimited.

SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Planning Activities



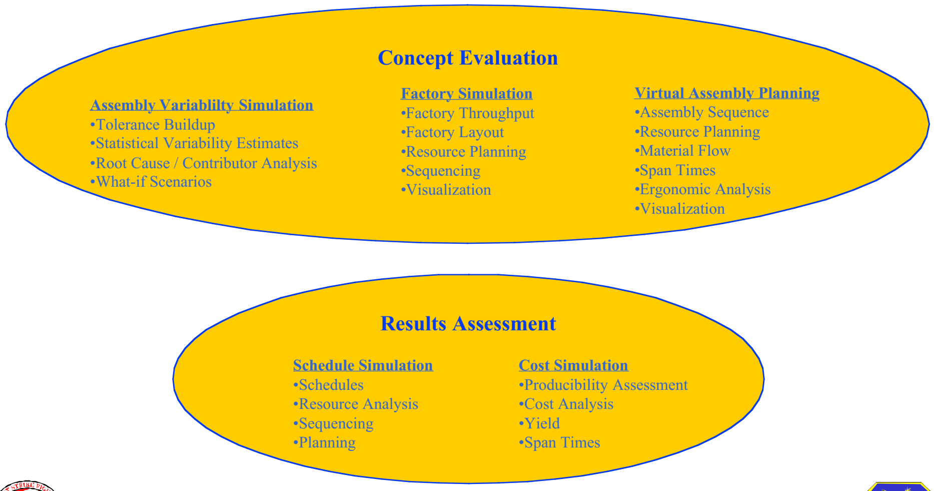
Distribution Statement A: Approved for public release. Distribution is unlimited.

SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Example Types of Simulations for SAVE



Concept Evaluation

Assembly Variablilty Simulation
- Tolerance Buildup
- Statistical Variability Estimates
- Root Cause / Contributor Analysis
- What-if Scenarios

Factory Simulation
- Factory Throughput
- Factory Layout
- Resource Planning
- Sequencing
- Visualization

Virtual Assembly Planning
- Assembly Sequence
- Resource Planning
- Material Flow
- Span Times
- Ergonomic Analysis
- Visualization

Results Assessment

Schedule Simulation
- Schedules
- Resource Analysis
- Sequencing
- Planning

Cost Simulation
- Producibility Assessment
- Cost Analysis
- Yield
- Span Times

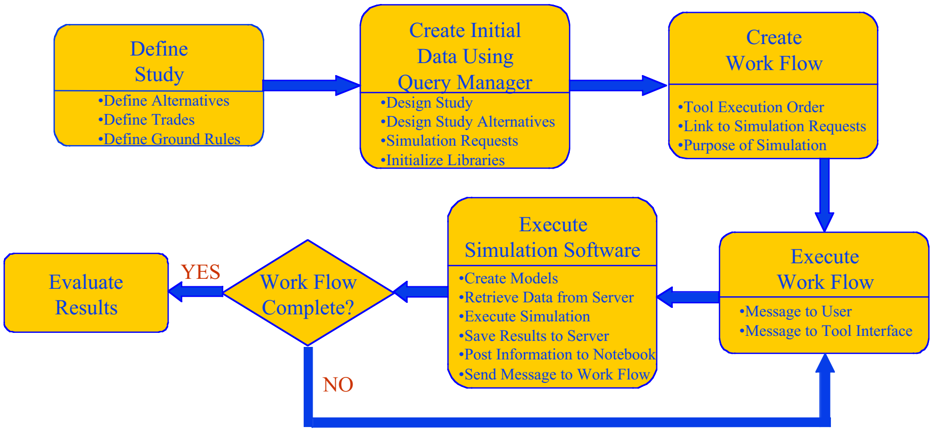
Distribution Statement A: Approved for public release. Distribution is unlimited.

SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Use of the SAVE Environment



Define Study
- Define Alternatives
- Define Trades
- Define Ground Rules

Create Initial Data Using Query Manager
- Design Study
- Design Study Alternatives
- Simulation Requests
- Initialize Libraries

Create Work Flow
- Tool Execution Order
- Link to Simulation Requests
- Purpose of Simulation

Execute Work Flow
- Message to User
- Message to Tool Interface

Execute Simulation Software
- Create Models
- Retrieve Data from Server
- Execute Simulation
- Save Results to Server
- Post Information to Notebook
- Send Message to Work Flow

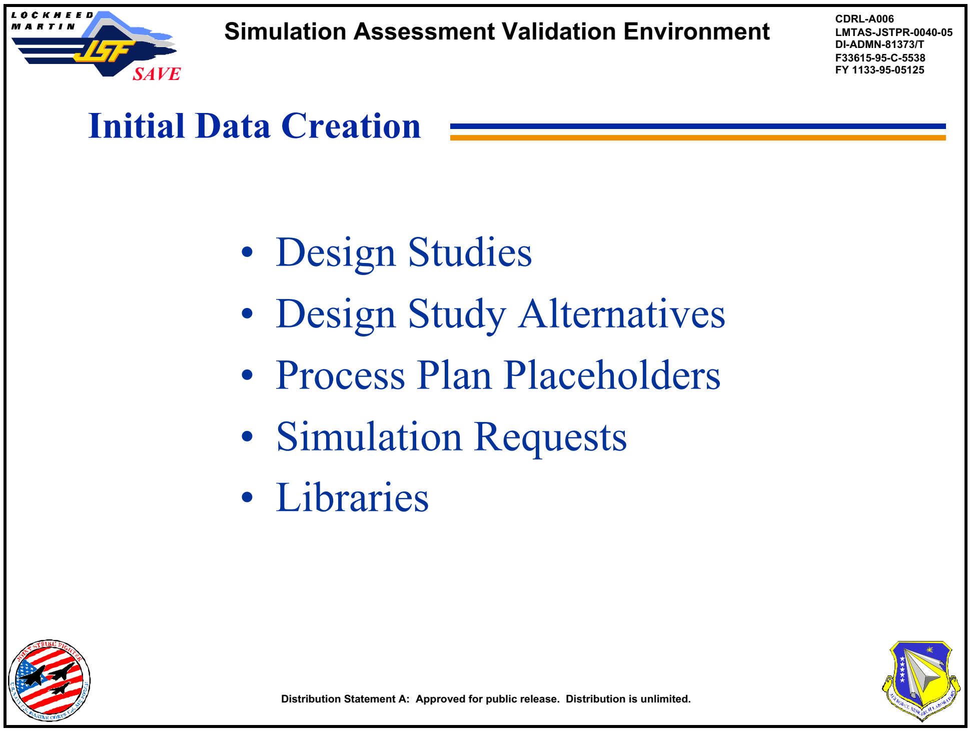
Work Flow Complete?

YES

NO

Evaluate Results

Distribution Statement A: Approved for public release. Distribution is unlimited.



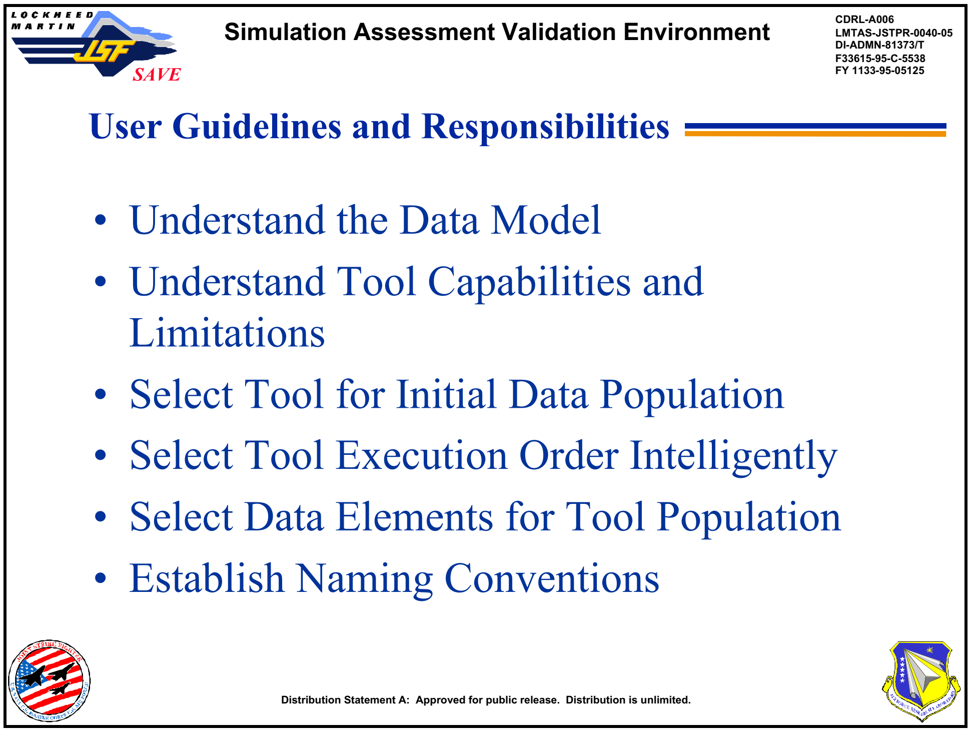
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Initial Data Creation

- Design Studies
- Design Study Alternatives
- Process Plan Placeholders
- Simulation Requests
- Libraries

Distribution Statement A: Approved for public release. Distribution is unlimited.



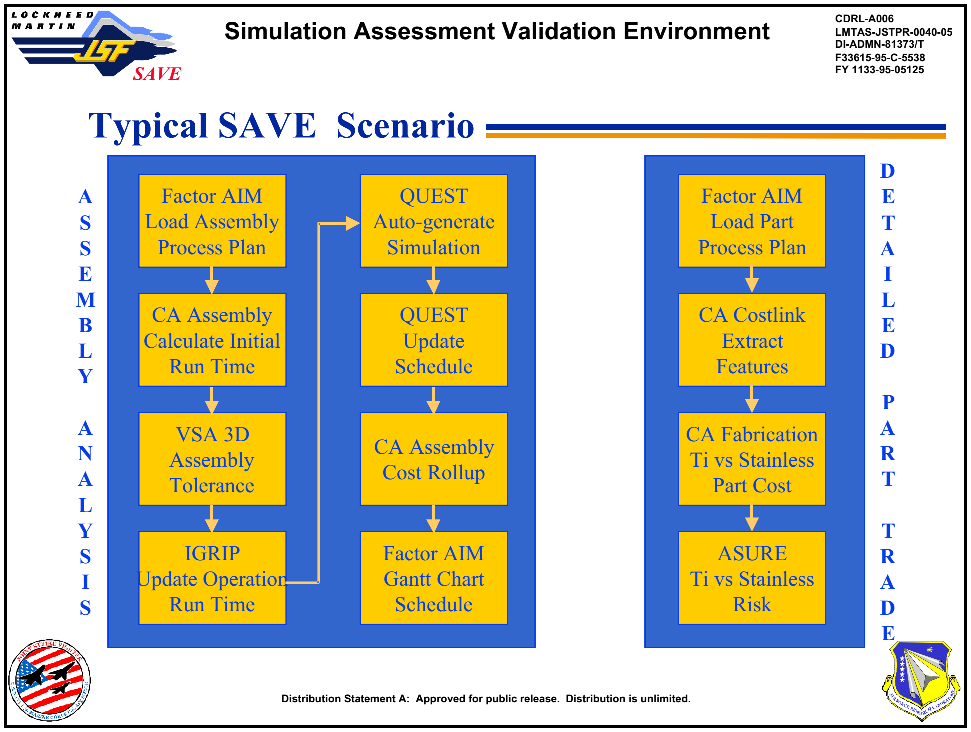
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

User Guidelines and Responsibilities

- Understand the Data Model
- Understand Tool Capabilities and Limitations
- Select Tool for Initial Data Population
- Select Tool Execution Order Intelligently
- Select Data Elements for Tool Population
- Establish Naming Conventions

Distribution Statement A: Approved for public release. Distribution is unlimited.



Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Typical SAVE Scenario

ASSEMBLY ANALYSIS

Factor AIM Load Assembly Process Plan

CA Assembly Calculate Initial Run Time

VSA 3D Assembly Tolerance

IGRIP Update Operation Run Time

QUEST Auto-generate Simulation

QUEST Update Schedule

CA Assembly Cost Rollup

Factor AIM Gantt Chart Schedule

DETAILED PART TRADE

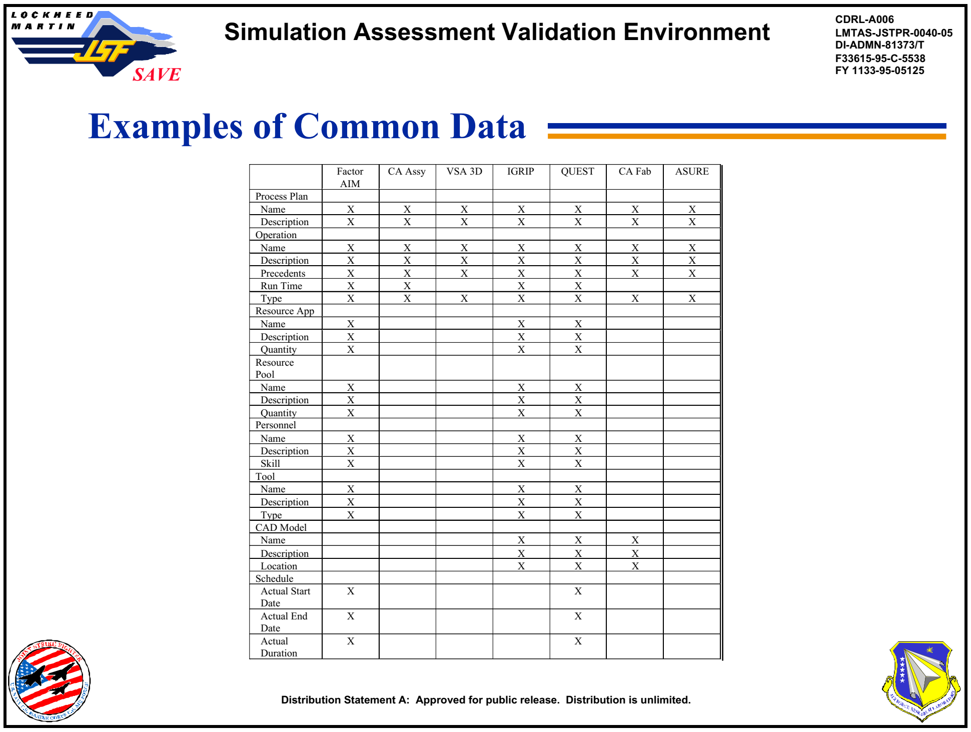
Factor AIM Load Part Process Plan

CA Costlink Extract Features

CA Fabrication Ti vs Stainless Part Cost

ASURE Ti vs Stainless Risk

Distribution Statement A: Approved for public release. Distribution is unlimited.



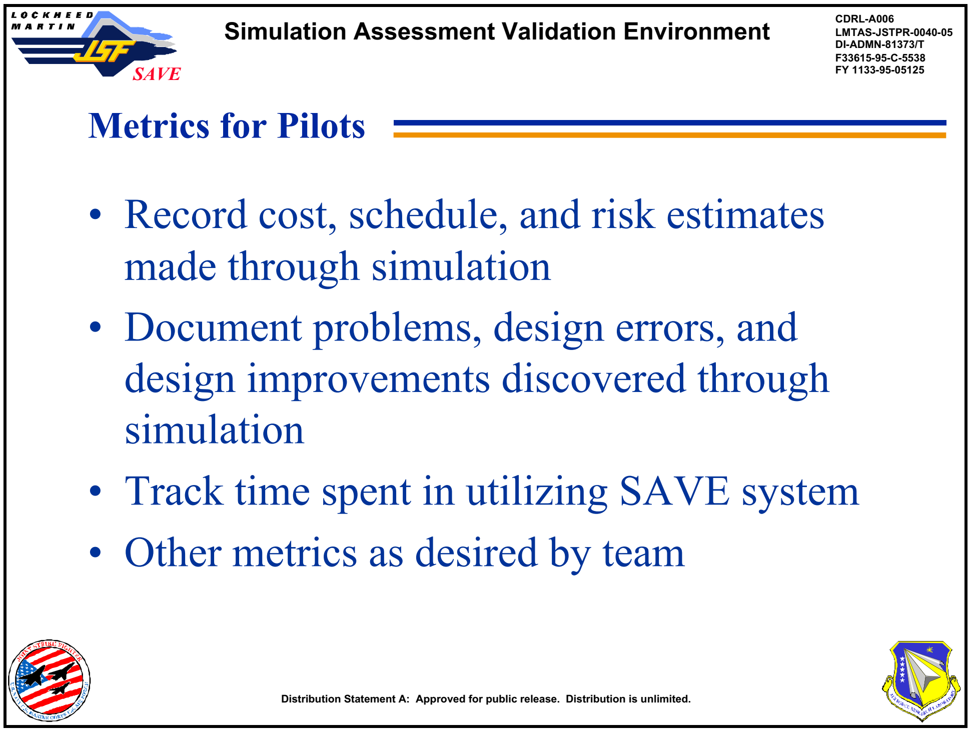
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Examples of Common Data

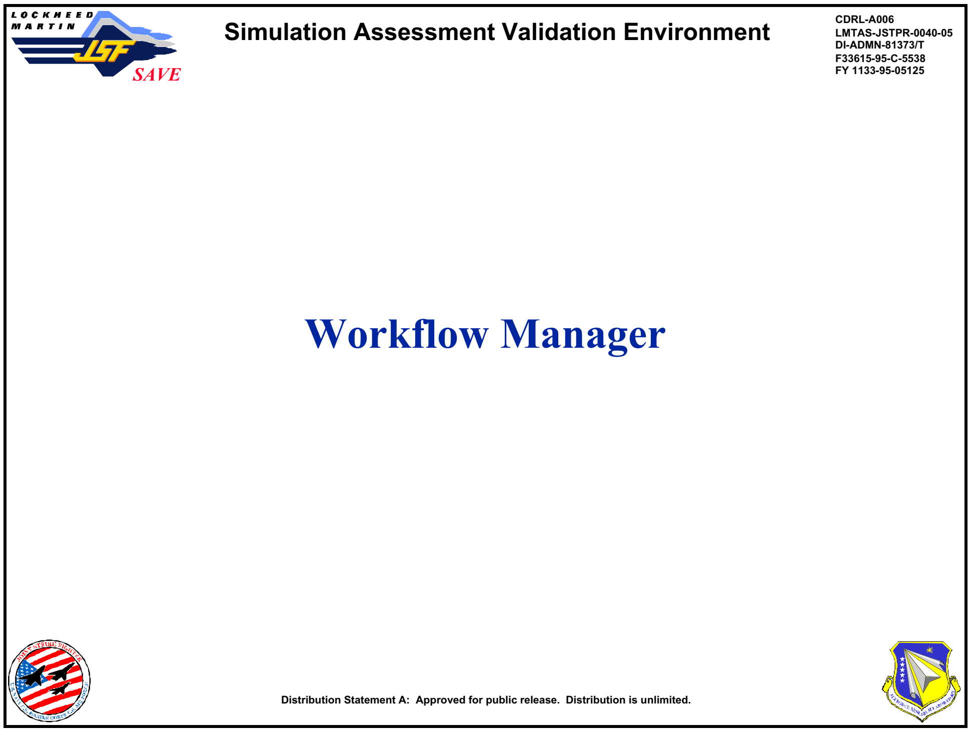
	Factor AIM	CA Assy	VSA 3D	IGRIP	QUEST	CA Fab	ASURE
Process Plan							
Name	X	X	X	X	X	X	X
Description	X	X	X	X	X	X	X
Operation							
Name	X	X	X	X	X	X	X
Description	X	X	X	X	X	X	X
Precedents	X	X	X	X	X	X	X
Run Time	X	X		X	X		
Type	X	X	X	X	X	X	X
Resource App							
Name	X			X	X		
Description	X			X	X		
Quantity	X			X	X		
Resource Pool							
Name	X			X	X		
Description	X			X	X		
Quantity	X			X	X		
Personnel							
Name	X			X	X		
Description	X			X	X		
Skill	X			X	X		
Tool							
Name	X			X	X		
Description	X			X	X		
Type	X			X	X		
CAD Model							
Name				X	X	X	
Description				X	X	X	
Location				X	X	X	
Schedule							
Actual Start Date	X				X		
Actual End Date	X				X		
Actual Duration	X				X		

Distribution Statement A: Approved for public release. Distribution is unlimited.



Metrics for Pilots

- Record cost, schedule, and risk estimates made through simulation
- Document problems, design errors, and design improvements discovered through simulation
- Track time spent in utilizing SAVE system
- Other metrics as desired by team



Workflow Manager

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Workflow Manager

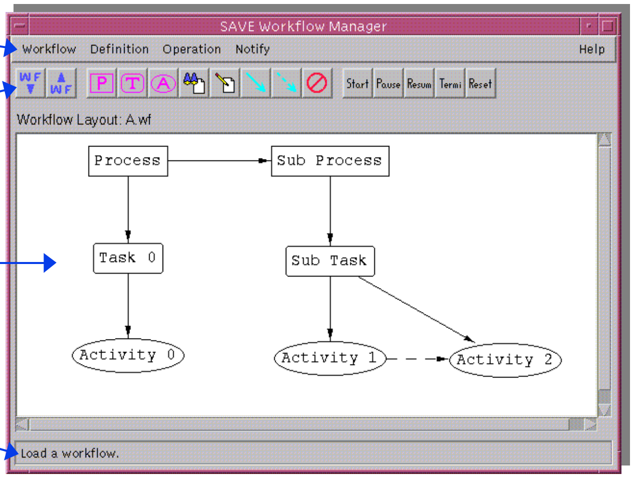
- Graphically models a process with decomposition down to the activity level
- Defines dependency relationships among components of the process
- Executes the process
  - Sends messages to users and tools
  - Monitors progress/status
  - Provides graphical feedback
- Implemented in JAVA for platform independence
- Communicates with simulation tools via CORBA

Distribution Statement A: Approved for public release. Distribution is unlimited.

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Workflow Manager Interface



Workflow Layout: A.wf

Load a workflow.

Menubar

Toolbar

Layout Area

Status Area

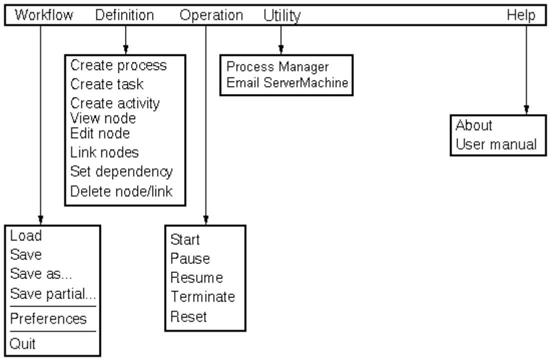
Distribution Statement A: Approved for public release. Distribution is unlimited.

SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Workflow Manager Menubar



- **Workflow** - Allows user to load or save a particular workflow model
- **Definition** - Allows user to create a workflow model with nodes and dependencies
- **Operation** - Allows user to execute the workflow mode
- **Notify** - Allows workflow manager to notify users via e-mail
- **Help** - Contains information about the use of the application

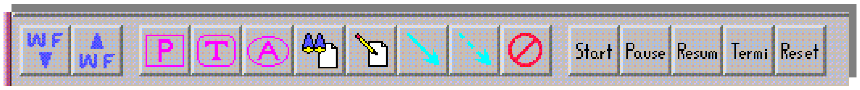
Distribution Statement A: Approved for public release. Distribution is unlimited.

SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Workflow Manager Toolbar



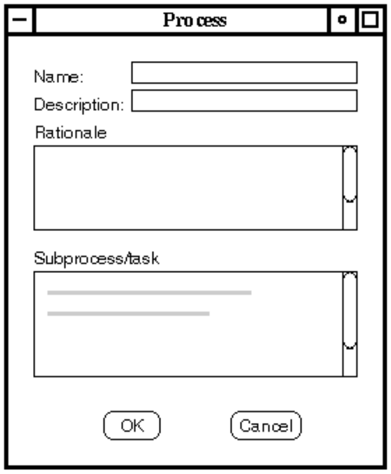
Workflow	Definition	Definition	Operation
Load	Create Process	View Node	Start
Save	Create Task	Edit Node	Pause
	Create Activity	Link Nodes	Resume
		Set Dependency	Terminate
		Delete Node/Link	Reset

Distribution Statement A: Approved for public release. Distribution is unlimited.

Creating a Workflow

- Two Primary Steps
  - Create Nodes (Process, Task, Activity)
  - Create Links (Model, Dependency)
- Creating Nodes
  - Select Type from Toolbar or Menubar
  - Click on the Layout to Place the Node
- Creating Links
  - Select Predecessor then Successor for Model
  - Select Source then Dependent for Dependency

Process Template



Process

Name:

Description:

Rationale

Subprocess/task

OK

Cancel

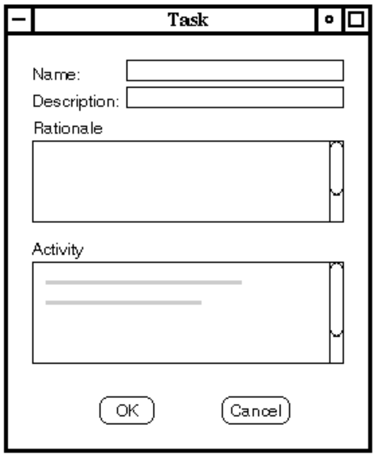
- Top Level Node in Workflow Model
- Supports One Predecessor Node (Process)
- Supports Multiple Successor Nodes (Process or Task)

LOCKHEED MARTIN JSF SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Task Template



- Container for Activities
- Supports One Predecessor Node (Process)
- Supports Multiple Successor Nodes (Activity)

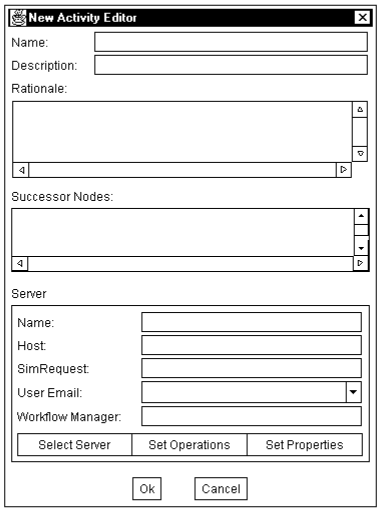
Distribution Statement A: Approved for public release. Distribution is unlimited.

LOCKHEED MARTIN JSF SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Activity Template



- Controls a Single Simulation Tool Wrapper
- Supports One Task or Multiple Activity Predecessor Nodes
- Supports Multiple Successor Nodes (Activity)
- Required Template Entries
  - Server Name
  - Server Host
  - SimRequest
  - Set Operations
  - Set Properties
  - User Email if Interactive Tool

Distribution Statement A: Approved for public release. Distribution is unlimited.

Running a Workflow

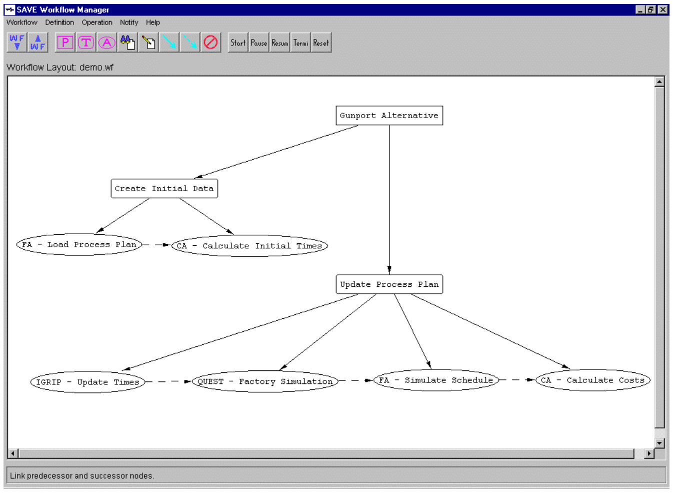
- Workflow Operations
  - Start
  - Pause
  - Resume
  - Terminate
  - Reset

• Select Operation from Menubar or Toolbar
• Click on Desired Node

- Workflow Status
  - White: Reset State
  - Blue: Initialized and Enabled State
  - Green: Working State
  - Yellow: Pause State
  - Gray: Completed State
  - Red: Terminate State
  - Orange with Red Outline: Faulty State

Distribution Statement A: Approved for public release. Distribution is unlimited.

Sample SAVE Workflow



Workflow Layout: demo.wf

Link predecessor and successor nodes.

Distribution Statement A: Approved for public release. Distribution is unlimited.

Query Manager

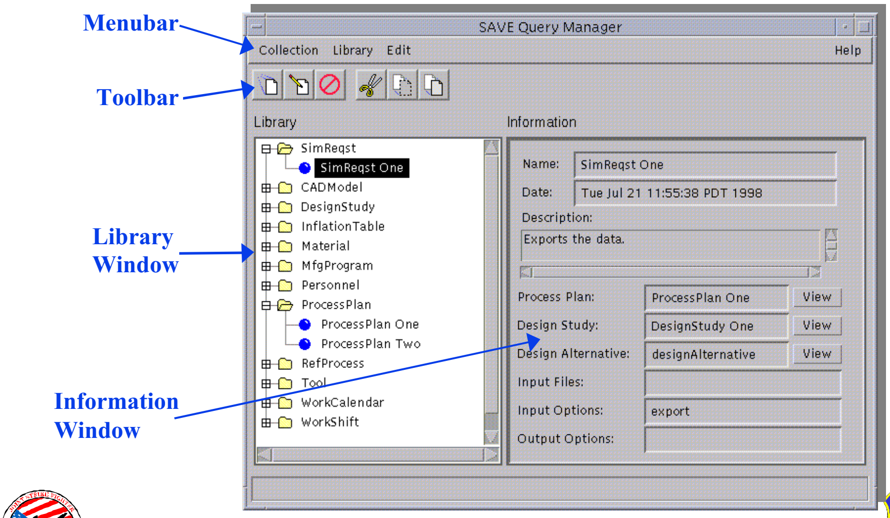
SAVE Query System

- Provides visibility into the SAVE data
- Provides capability to
  - Browse objects /attributes
  - Create objects /attributes
  - Modify objects /attributes
  - Delete objects /attributes
- Implemented in JAVA for platform independence
- Communicates with the SAVE data model via CORBA

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Query Manager Interface


Menubar

Toolbar

Library Window

Information Window

SAVE Query Manager

Library

- SimReqst
  - SimReqst One
- CADModel
- DesignStudy
- InflationTable
- Material
- MfgProgram
- Personnel
- ProcessPlan
  - ProcessPlan One
  - ProcessPlan Two
- RefProcess
- Tool
- WorkCalendar
- WorkShift

Information

Name: SimReqst One

Date: Tue Jul 21 11:55:38 PDT 1998

Description: Exports the data.

Process Plan: ProcessPlan One View

Design Study: DesignStudy One View

Design Alternative: designAlternative View

Input Files:

Input Options: export

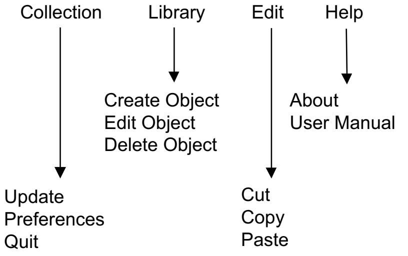
Output Options:

Distribution Statement A: Approved for public release. Distribution is unlimited.

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Query Manager Menubar


Collection → Update, Preferences, Quit

Library → Create Object, Edit Object, Delete Object

Edit → Cut, Copy, Paste

Help → About, User Manual

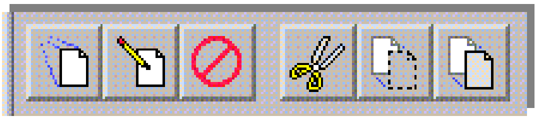
- **Collection** - Allows user to update information and set preferences for the QM
- **Library** - Allows user to create, edit, or delete SAVE library objects
- **Edit** - Allows user to cut, copy, or paste text
- **Help** - Contains information about the use of the application

Distribution Statement A: Approved for public release. Distribution is unlimited.

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Query Manager Toolbar



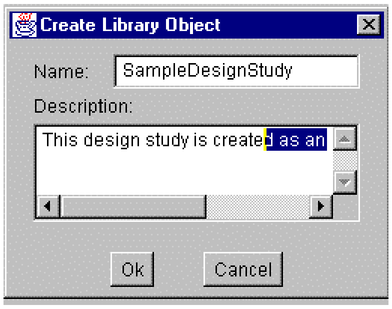
Library	Edit
Create Object	Cut
Edit Object	Copy
Delete Object	Paste

Distribution Statement A: Approved for public release. Distribution is unlimited.

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Creating Library Objects



Create Library Object

Name: SampleDesignStudy

Description:

This design study is created as an

- Select create object from menubar or toolbar
- Highlight the appropriate library in the library window
- Enter object name and description in popup window
- Click OK to create object in database
- Populate other attributes using edit function

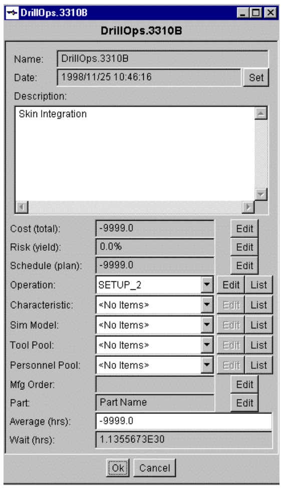
Distribution Statement A: Approved for public release. Distribution is unlimited.

LOCKHEED MARTIN JSF SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Editing Library Objects



DrillOps.3310B

Name: DrillOps.3310B
Date: 1998/11/25 10:46:16 Set
Description:
Skin Integration

Cost (total): -9999.0 Edit
Risk (yield): 0.0% Edit
Schedule (plan): -9999.0 Edit
Operation: SETUP_2 Edit List
Characteristic: <No Items> Edit List
Sim Model: <No Items> Edit List
Tool Pool: <No Items> Edit List
Personnel Pool: <No Items> Edit List
Mfg Order: Edit
Part: Part Name Edit
Average (hrs): -9999.0
Wait (hrs): 1.1355673E30

Ok Cancel

- Select edit object from menubar or toolbar
- Highlight the appropriate object instance in the library window
- Set current date/time stamp by selecting set button
- Edit simple attributes directly by modifying field
- Edit object attributes by clicking edit button
- List object sequences by clicking list button
- Click OK to save changes

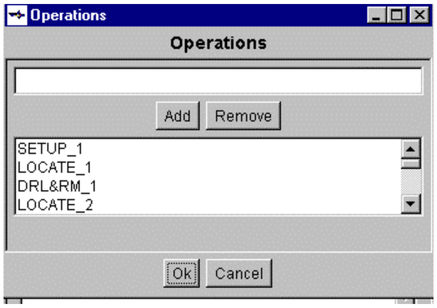
Distribution Statement A: Approved for public release. Distribution is unlimited.

LOCKHEED MARTIN JSF SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Object Sequence List



Operations

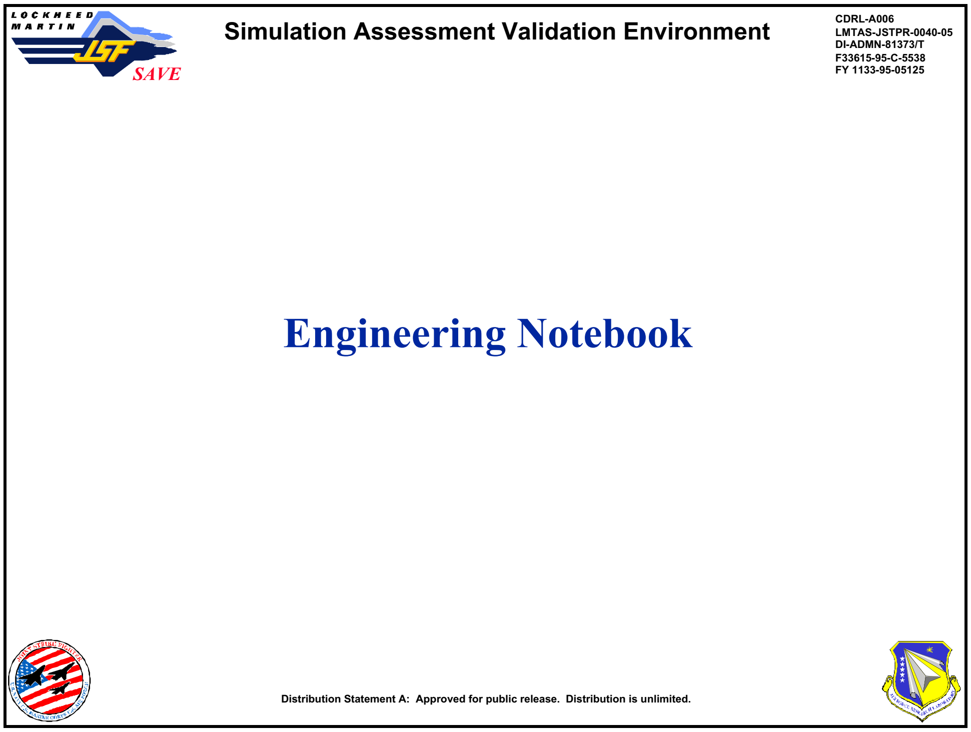
Add Remove

SETUP_1
LOCATE_1
DRL&RM_1
LOCATE_2

Ok Cancel

- Scroll list displays names of objects in sequence
- To view an object in the parent window, highlight in the list and click OK
- To remove object, highlight in list and click remove button
- To add object, click add button and enter information in create popup window
- Click OK to save changes

Distribution Statement A: Approved for public release. Distribution is unlimited.

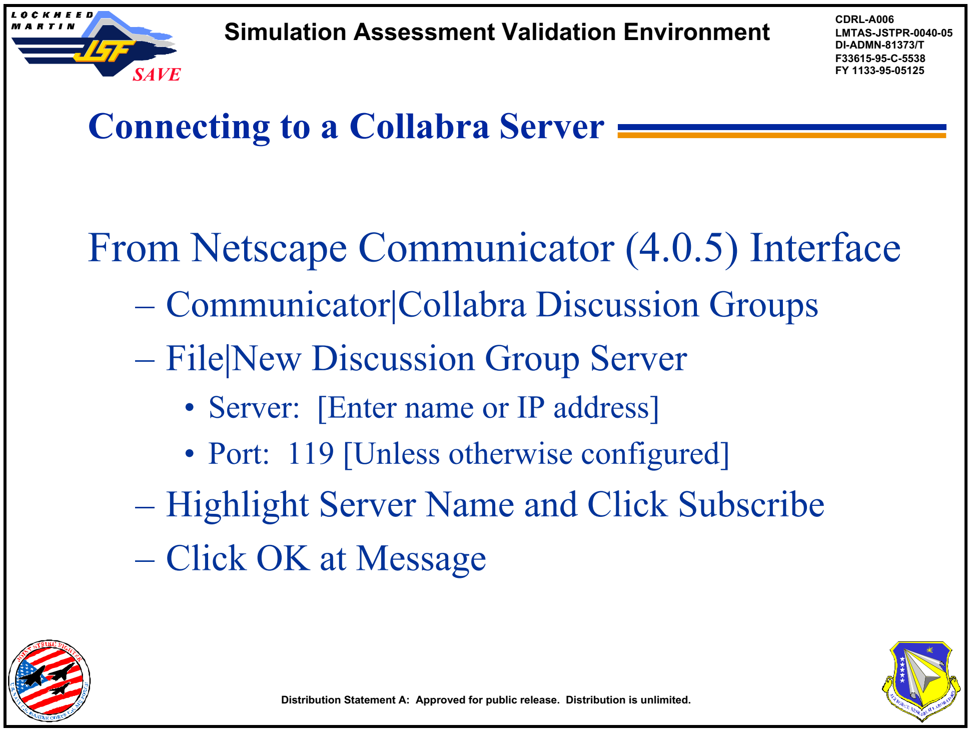


Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Engineering Notebook

Distribution Statement A: Approved for public release. Distribution is unlimited.



Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

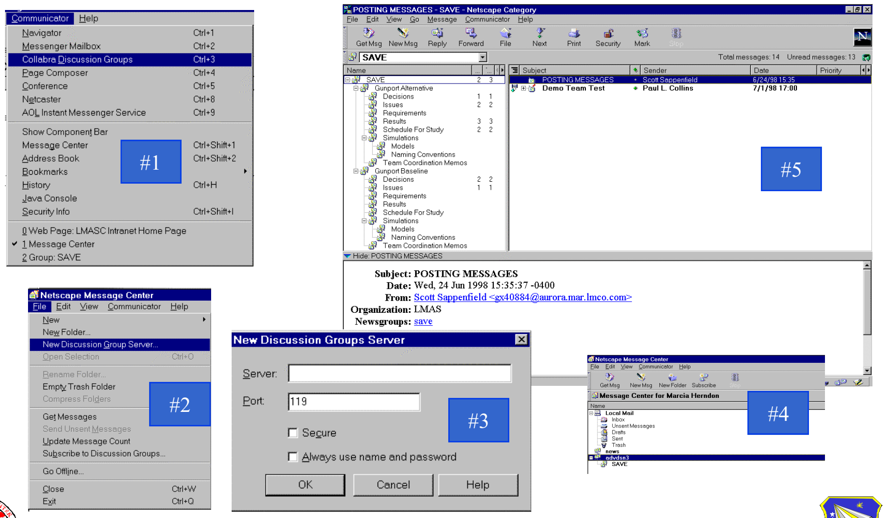
Connecting to a Collabra Server

From Netscape Communicator (4.0.5) Interface

- Communicator|Collabra Discussion Groups
- File|New Discussion Group Server
  - Server: [Enter name or IP address]
  - Port: 119 [Unless otherwise configured]
- Highlight Server Name and Click Subscribe
- Click OK at Message

Distribution Statement A: Approved for public release. Distribution is unlimited.

Collabra Interface



Server:

Port: 119

Total messages: 14 Unread messages: 13

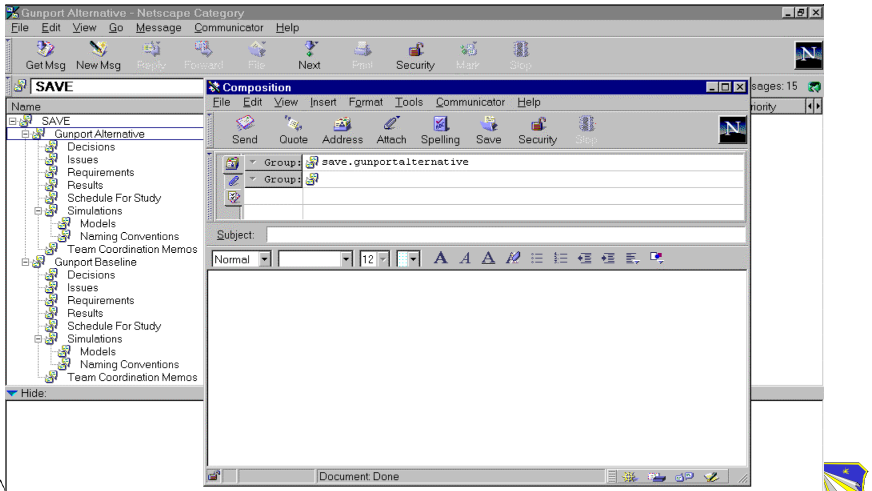
Subject	Sender	Date	Priority
POSTING MESSAGES	Scott Sappenfield	6/24/98 15:35	
Demo Team Test	Paul L. Collins	7/1/98 17:00	

Subject: POSTING MESSAGES
Date: Wed, 24 Jun 1998 15:35:37 -0400
From: Scott Sappenfield <gx40884@aurora.mar.lmco.com>
Organization: LMAS
Newsgroups: save

Posting Messages to Discussion Groups

- From Collabra Discussion Group Menu
  - Double Click on Name of Group
  - Choose Appropriate Subgroup for Posting
  - Click NewMsg on Toolbar
  - Type Message in Composition Window
- Messages are Threaded Like Internet NewsGroups

Message Posting Interface



Group: save.gunportalternative
Group:
Subject:

Document: Done

Distribution Statement A: Approved for public release. Distribution is unlimited.

Sample Discussion Group Hierarchy

- Decisions
- Issues
- Requirements
- Results
- Schedule
- Simulations
- Coordination Memos

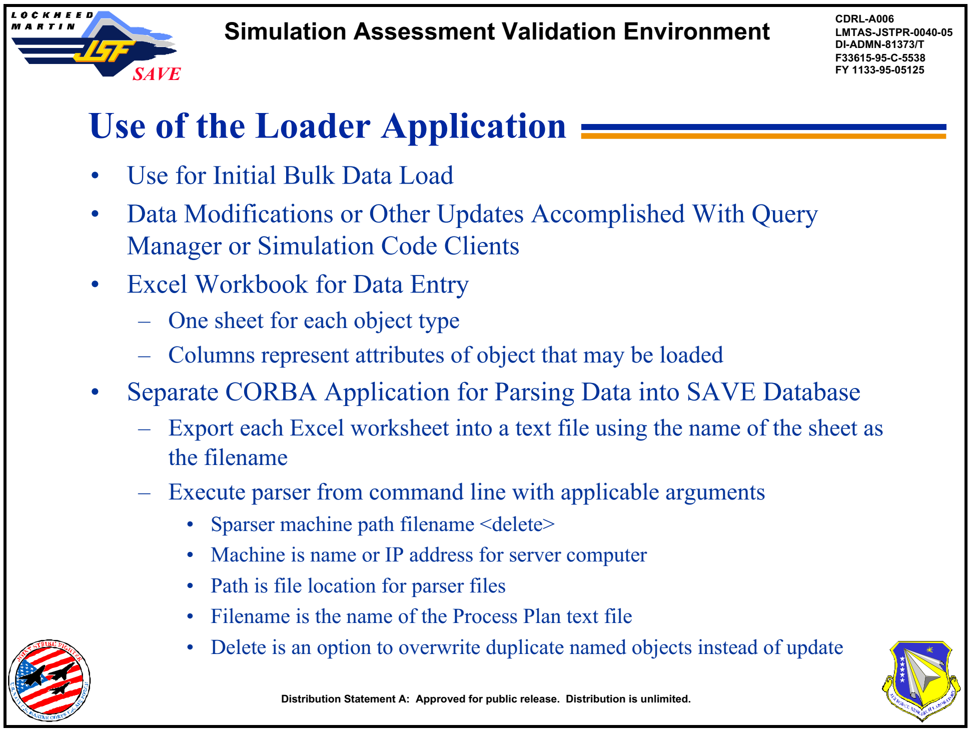
Distribution Statement A: Approved for public release. Distribution is unlimited.

Connecting to a Collabra Server

From Netscape Communicator (4.0.5) Interface

- Communicator|Collabra Discussion Groups
- File|New Discussion Group Server
  - Server: [Enter name or IP address]
  - Port: 119 [Unless otherwise configured]

Loader Overview



LOCKHEED MARTIN JSF SAVE

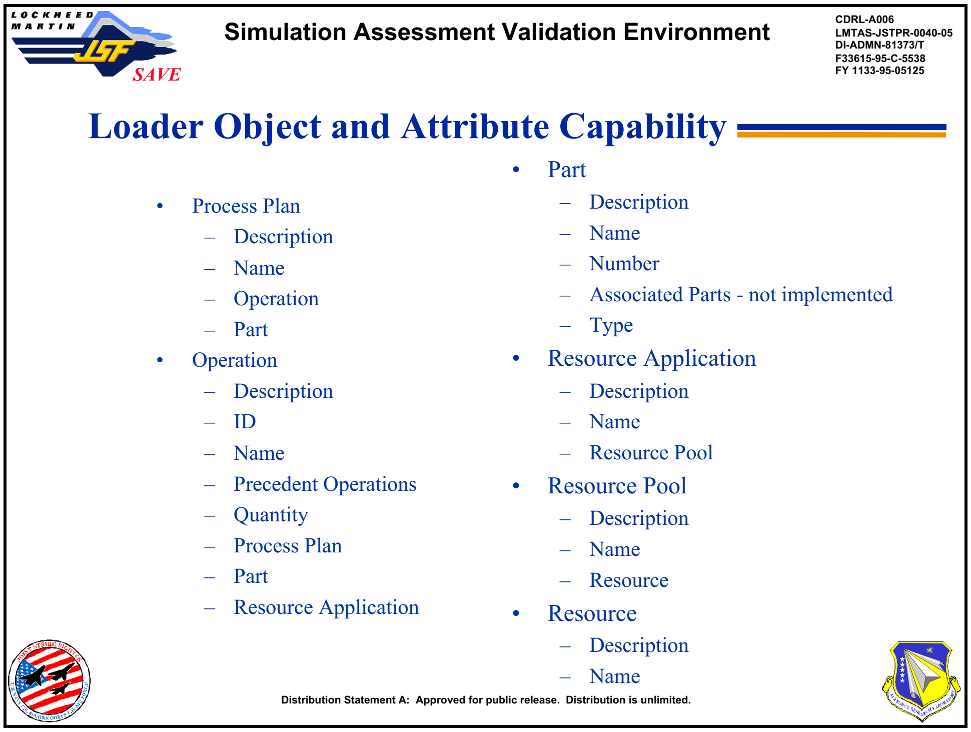
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Use of the Loader Application

- Use for Initial Bulk Data Load
- Data Modifications or Other Updates Accomplished With Query Manager or Simulation Code Clients
- Excel Workbook for Data Entry
  - One sheet for each object type
  - Columns represent attributes of object that may be loaded
- Separate CORBA Application for Parsing Data into SAVE Database
  - Export each Excel worksheet into a text file using the name of the sheet as the filename
  - Execute parser from command line with applicable arguments
    - Sparser machine path filename <delete>
    - Machine is name or IP address for server computer
    - Path is file location for parser files
    - Filename is the name of the Process Plan text file
    - Delete is an option to overwrite duplicate named objects instead of update

Distribution Statement A: Approved for public release. Distribution is unlimited.



LOCKHEED MARTIN JSF SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Loader Object and Attribute Capability

- Process Plan
  - Description
  - Name
  - Operation
  - Part
- Operation
  - Description
  - ID
  - Name
  - Precedent Operations
  - Quantity
  - Process Plan
  - Part
  - Resource Application
- Part
  - Description
  - Name
  - Number
  - Associated Parts - not implemented
  - Type
- Resource Application
  - Description
  - Name
  - Resource Pool
- Resource Pool
  - Description
  - Name
  - Resource
- Resource
  - Description
  - Name

Distribution Statement A: Approved for public release. Distribution is unlimited.

Sample Excel Worksheet for Operations

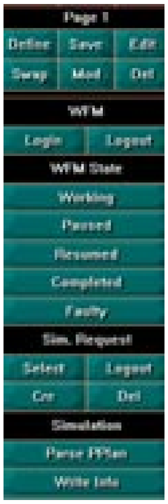
Operation ID	Name	Description	Process Plan	Precedent Operations
op1	LoadFixture1	Position parts in fixture	-	-
op2	DrillReam1	Drill holes in parts	-	op1
op3	AssembleBox	Wing box assembly	pp2	op1 op2
op4	Setup1	Setup tooling for wing box	-	-
op5	LoadFixture2	Position parts in fixture	-	op4
op6	DrillReam2	Drill holes in parts	-	op4 op5

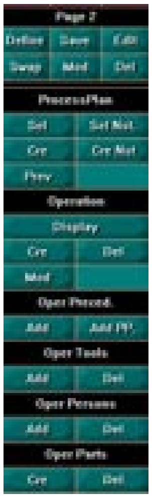
IGRIP/QUEST Wrapper Overview

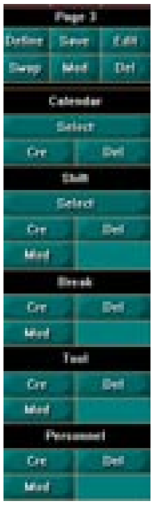
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

User Pages for SAVE-Specific Items







Distribution Statement A: Approved for public release. Distribution is unlimited.

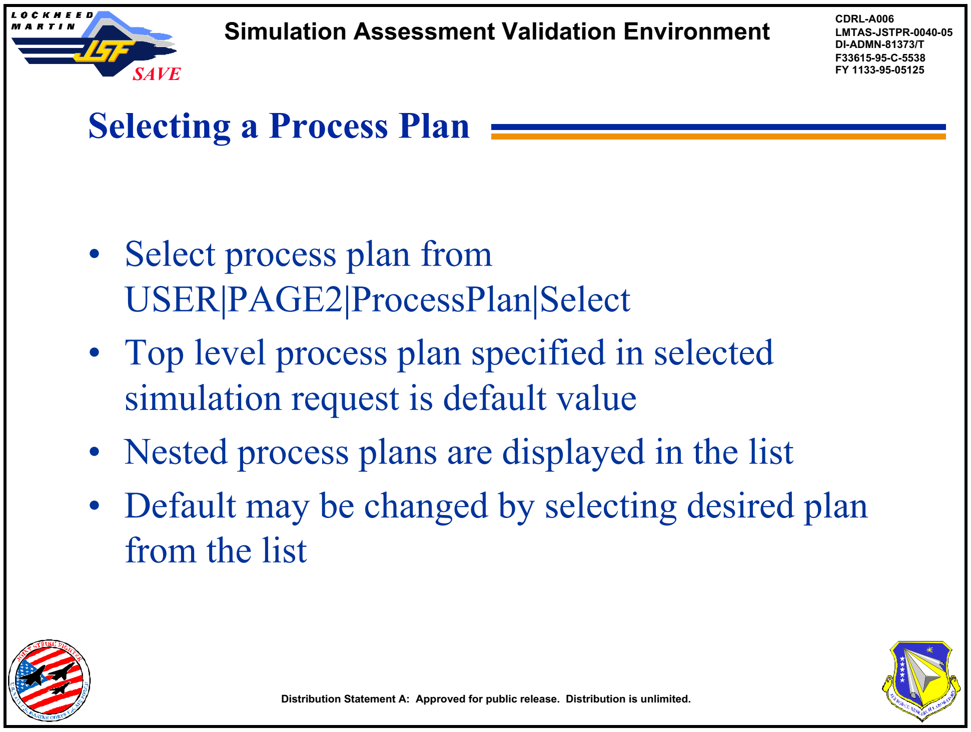
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Connections to SAVE Server

- Connecting
  - Select simulation request from USER|PAGE1|Sim.Request|Select
  - Complete dialog box
    - Specify server name
    - Specify simulation request name
    - Defaults are from save.cfg file
    - Select “done” to establish connection
- Disconnecting
  - Select logout from USER|PAGE1|Sim.Request|Logout
  - User must logout to ensure that all database transactions are properly committed

Distribution Statement A: Approved for public release. Distribution is unlimited.



SAVE

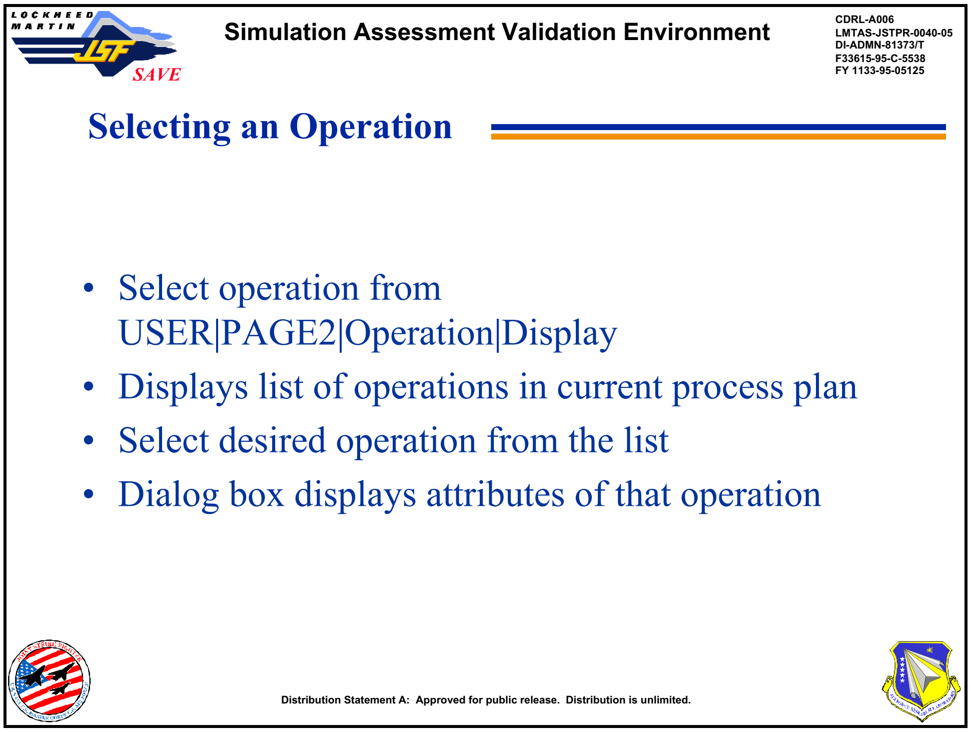
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Selecting a Process Plan

- Select process plan from USER|PAGE2|ProcessPlan|Select
- Top level process plan specified in selected simulation request is default value
- Nested process plans are displayed in the list
- Default may be changed by selecting desired plan from the list

Distribution Statement A: Approved for public release. Distribution is unlimited.



SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Selecting an Operation

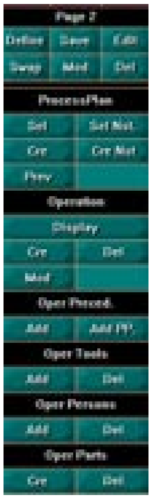
- Select operation from USER|PAGE2|Operation|Display
- Displays list of operations in current process plan
- Select desired operation from the list
- Dialog box displays attributes of that operation

Distribution Statement A: Approved for public release. Distribution is unlimited.

Establishing a Connection to the WFM

- Connect to Simulator using USER|PAGE1|WFM|Login
  - Informs the simulator server that QUEST or IGRIP is up and running and may accept messages from the WFM
  - Allows parameters to be passed from the WFM to QUEST or IGRIP via the simulator server
- Use WFM State buttons to communicate information to the WFM
  - Working
  - Paused
  - Resumed
  - Completed
  - Faulty
- Logout from Simulator using USER|PAGE1|WFM|Logout

User Function Descriptions - Page 2



Select process plans
Create new process plans
Select nested process plans
Create nested process plans
Select parent process plan

Display an operation
Create a new operation in active process plan
Delete operation
Modify operation runtime information - IGRIP only

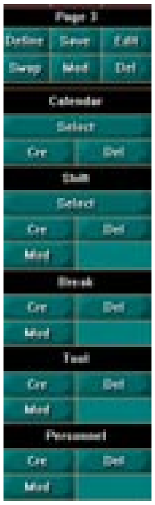
Add precedent operations from current process plan - QUEST only
Add precedent operations from another process plan - QUEST only

Add tools to an operation - QUEST only
Delete tools from an operation - QUEST only

Add personnel to an operation - QUEST only
Delete personnel from an operation - QUEST only

Create/update parts for an operation
Delete parts for an operation

User Function Descriptions - Page 3



Select calendars
Create calendars
Delete calendars

Select from a list of available shifts as the active one
Create a shift with start and end time
Modify the shift parameters
Delete the shift

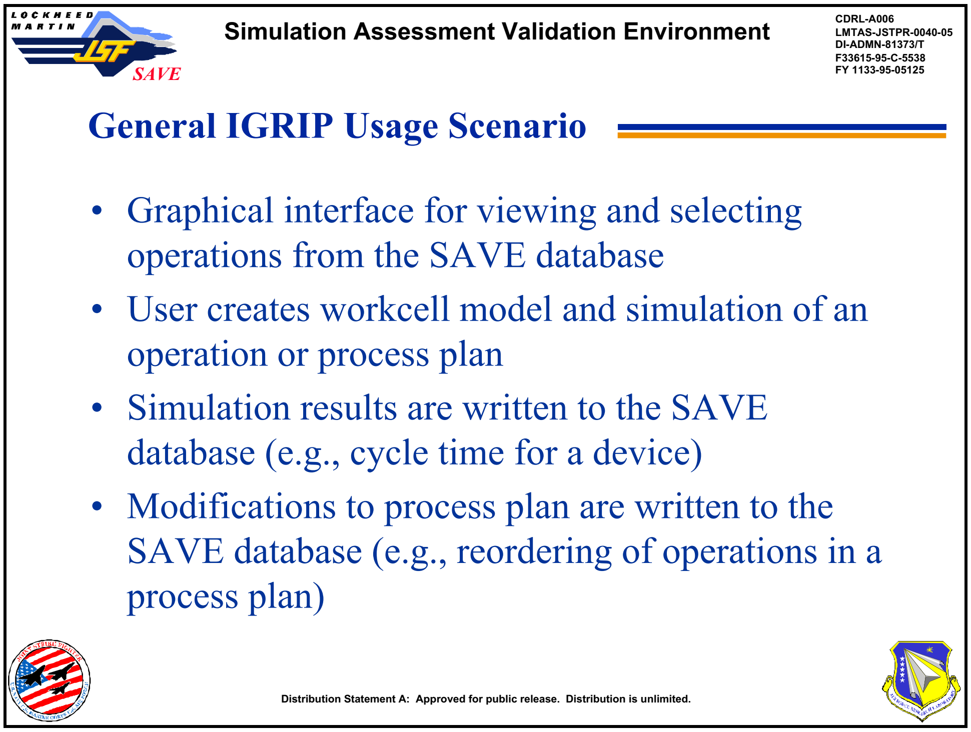
Add a break to the active shift
Modify an existing break
Delete a break

Create a new machine tool pool in the selected process plan
Modify an existing tool pool
Delete a tool pool

Create new personnel pool
Modify an existing personnel pool
Delete personnel

General IGRIP Usage Scenario

- Graphical interface for viewing and selecting operations from the SAVE database
- User creates workcell model and simulation of an operation or process plan
- Simulation results are written to the SAVE database (e.g., cycle time for a device)
- Modifications to process plan are written to the SAVE database (e.g., reordering of operations in a process plan)



LOCKHEED MARTIN
JSF
SAVE

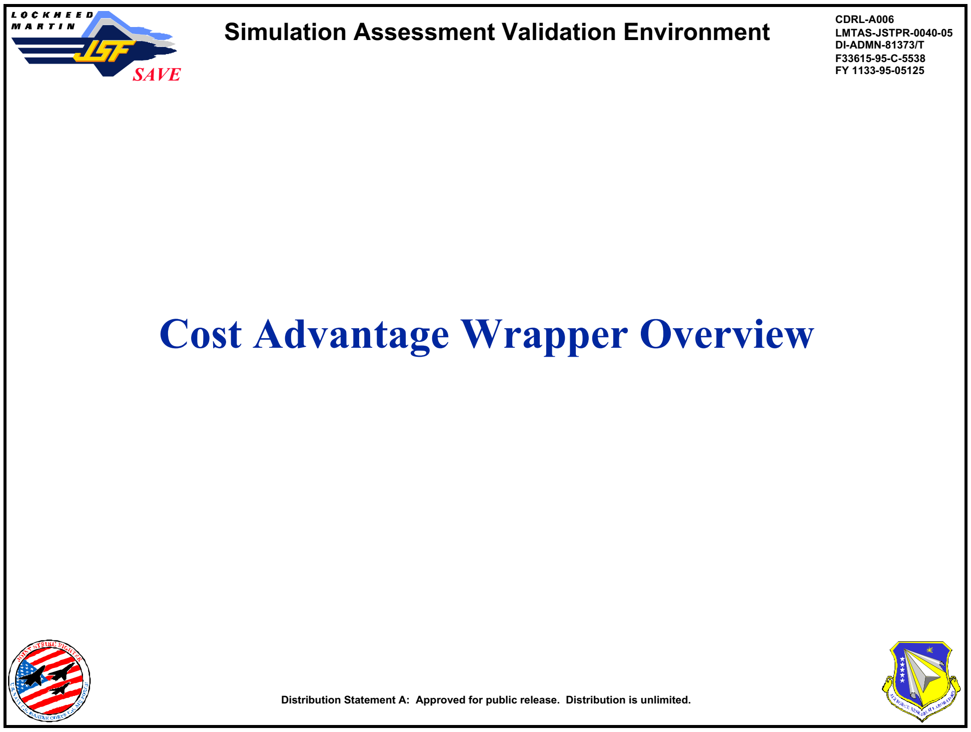
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

General IGRIP Usage Scenario

- Graphical interface for viewing and selecting operations from the SAVE database
- User creates workcell model and simulation of an operation or process plan
- Simulation results are written to the SAVE database (e.g., cycle time for a device)
- Modifications to process plan are written to the SAVE database (e.g., reordering of operations in a process plan)

Distribution Statement A: Approved for public release. Distribution is unlimited.



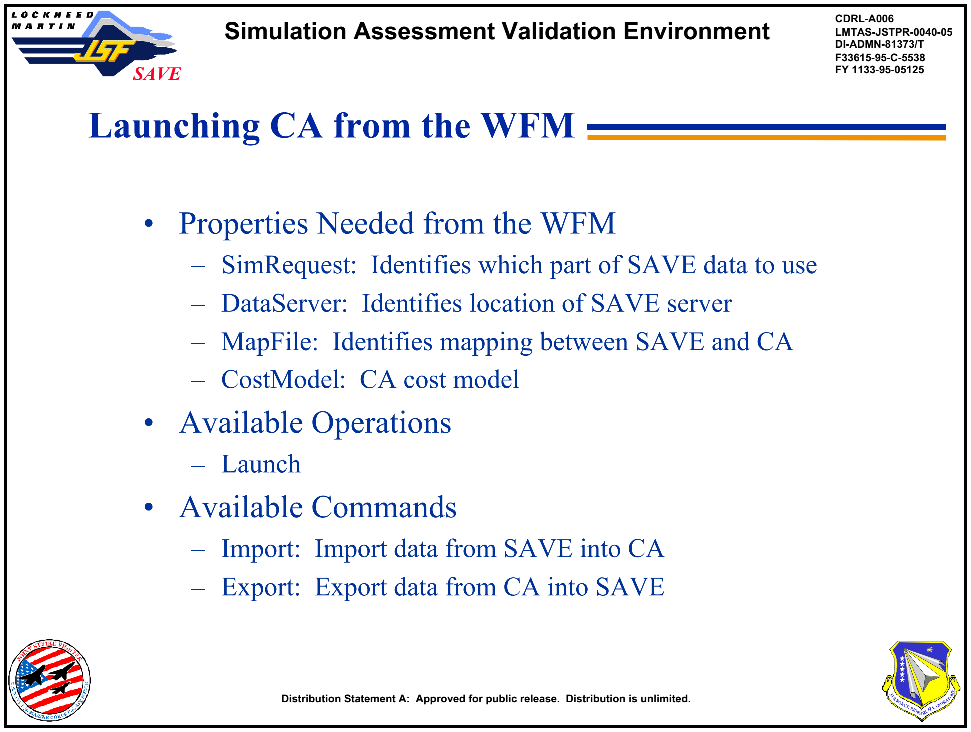
LOCKHEED MARTIN
JSF
SAVE

Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Cost Advantage Wrapper Overview

Distribution Statement A: Approved for public release. Distribution is unlimited.



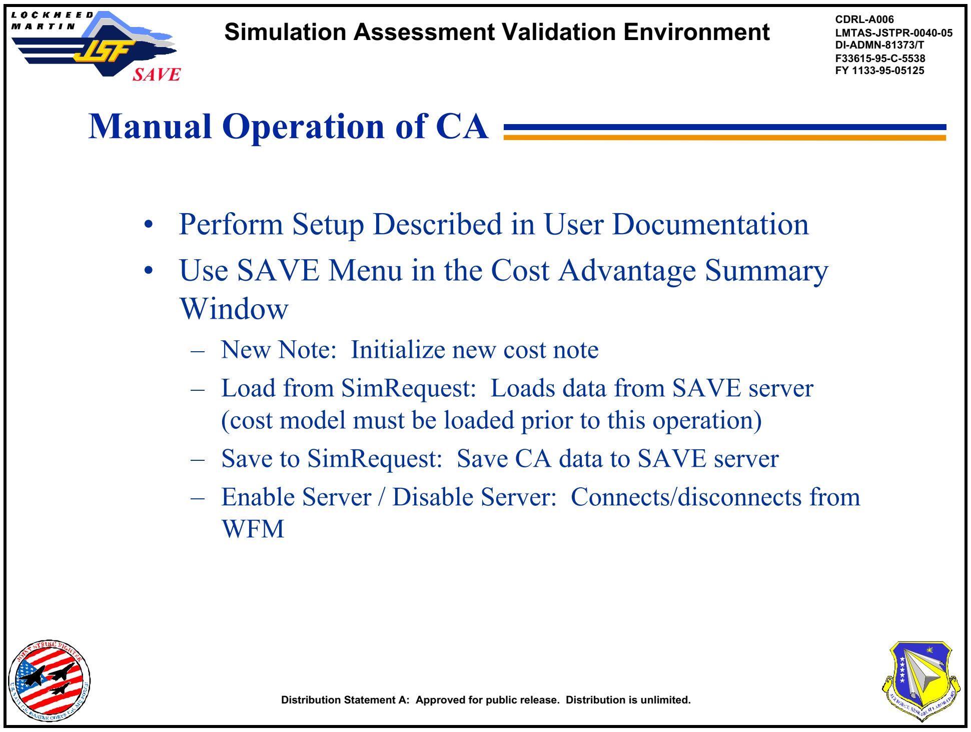
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Launching CA from the WFM

- Properties Needed from the WFM
  - SimRequest: Identifies which part of SAVE data to use
  - DataServer: Identifies location of SAVE server
  - MapFile: Identifies mapping between SAVE and CA
  - CostModel: CA cost model
- Available Operations
  - Launch
- Available Commands
  - Import: Import data from SAVE into CA
  - Export: Export data from CA into SAVE

Distribution Statement A: Approved for public release. Distribution is unlimited.



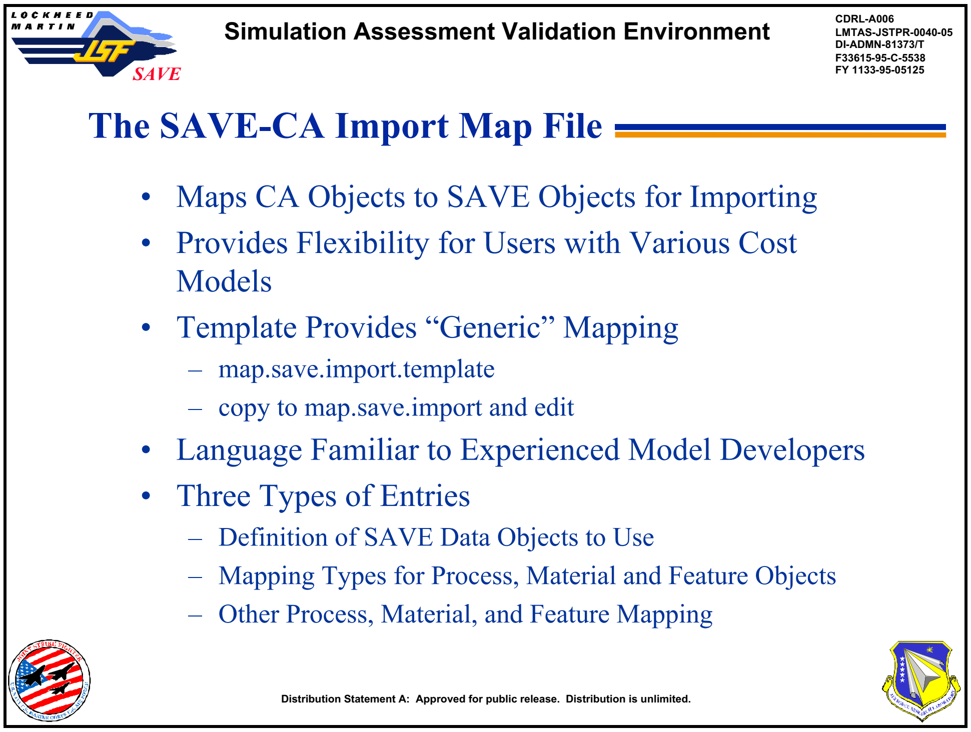
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Manual Operation of CA

- Perform Setup Described in User Documentation
- Use SAVE Menu in the Cost Advantage Summary Window
  - New Note: Initialize new cost note
  - Load from SimRequest: Loads data from SAVE server (cost model must be loaded prior to this operation)
  - Save to SimRequest: Save CA data to SAVE server
  - Enable Server / Disable Server: Connects/disconnects from WFM

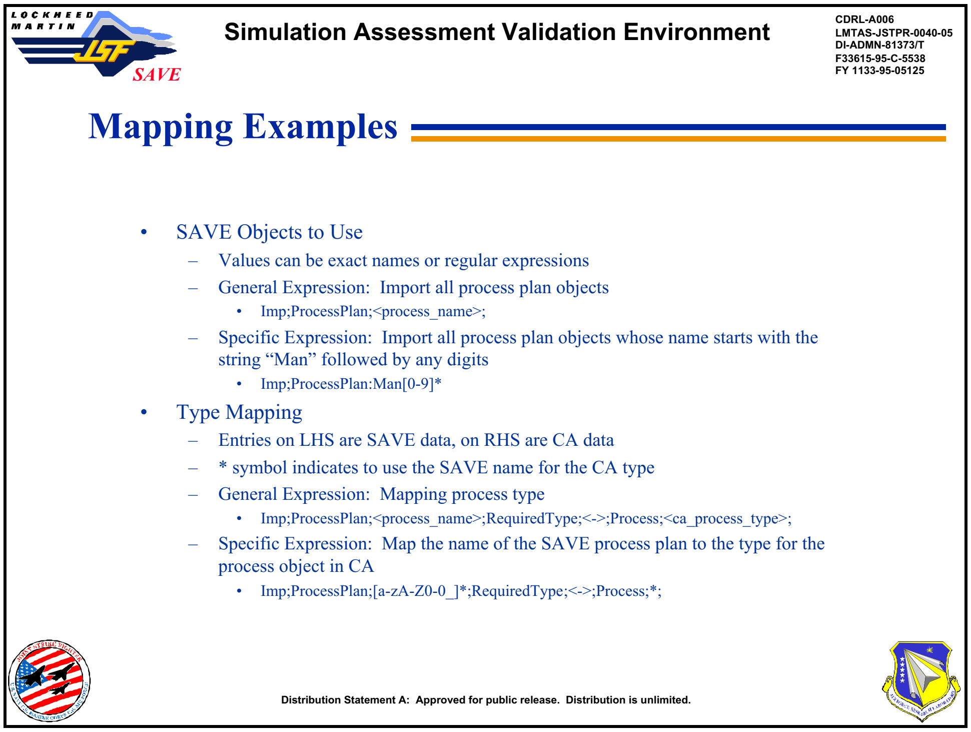
Distribution Statement A: Approved for public release. Distribution is unlimited.



The SAVE-CA Import Map File

- Maps CA Objects to SAVE Objects for Importing
- Provides Flexibility for Users with Various Cost Models
- Template Provides "Generic" Mapping
  - map.save.import.template
  - copy to map.save.import and edit
- Language Familiar to Experienced Model Developers
- Three Types of Entries
  - Definition of SAVE Data Objects to Use
  - Mapping Types for Process, Material and Feature Objects
  - Other Process, Material, and Feature Mapping

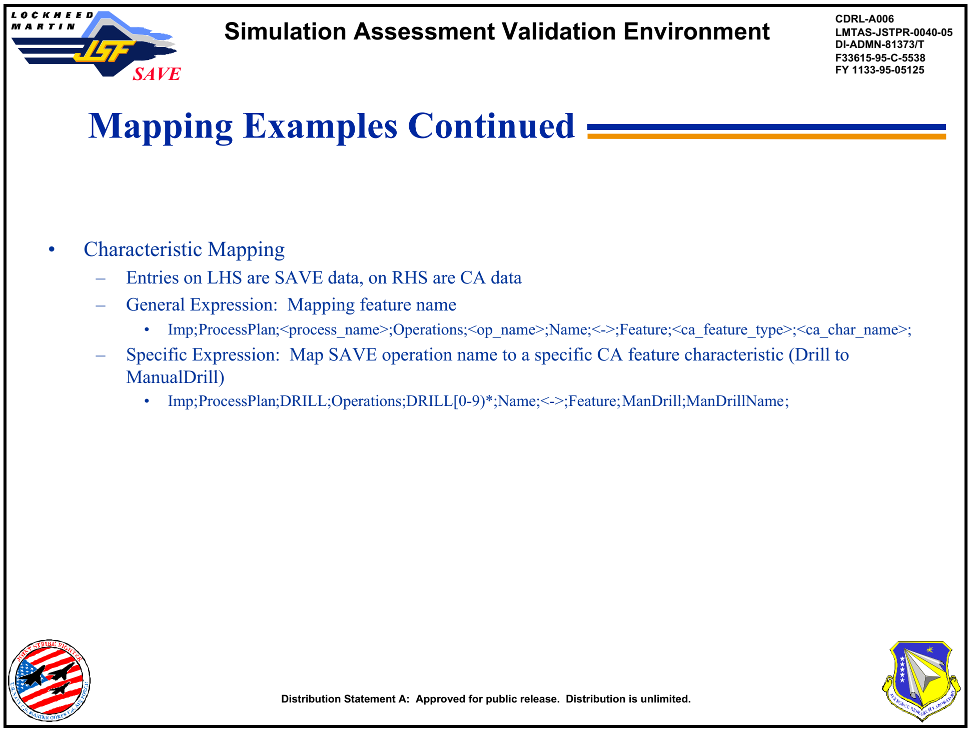
Distribution Statement A: Approved for public release. Distribution is unlimited.



Mapping Examples

- SAVE Objects to Use
  - Values can be exact names or regular expressions
  - General Expression: Import all process plan objects
    - Imp;ProcessPlan;<process_name>;
  - Specific Expression: Import all process plan objects whose name starts with the string "Man" followed by any digits
    - Imp;ProcessPlan:Man[0-9]*
- Type Mapping
  - Entries on LHS are SAVE data, on RHS are CA data
  - * symbol indicates to use the SAVE name for the CA type
  - General Expression: Mapping process type
    - Imp;ProcessPlan;<process_name>;RequiredType;<->;Process;<ca_process_type>;
  - Specific Expression: Map the name of the SAVE process plan to the type for the process object in CA
    - Imp;ProcessPlan;[a-zA-Z0-0_]*;RequiredType;<->;Process;*;

Distribution Statement A: Approved for public release. Distribution is unlimited.



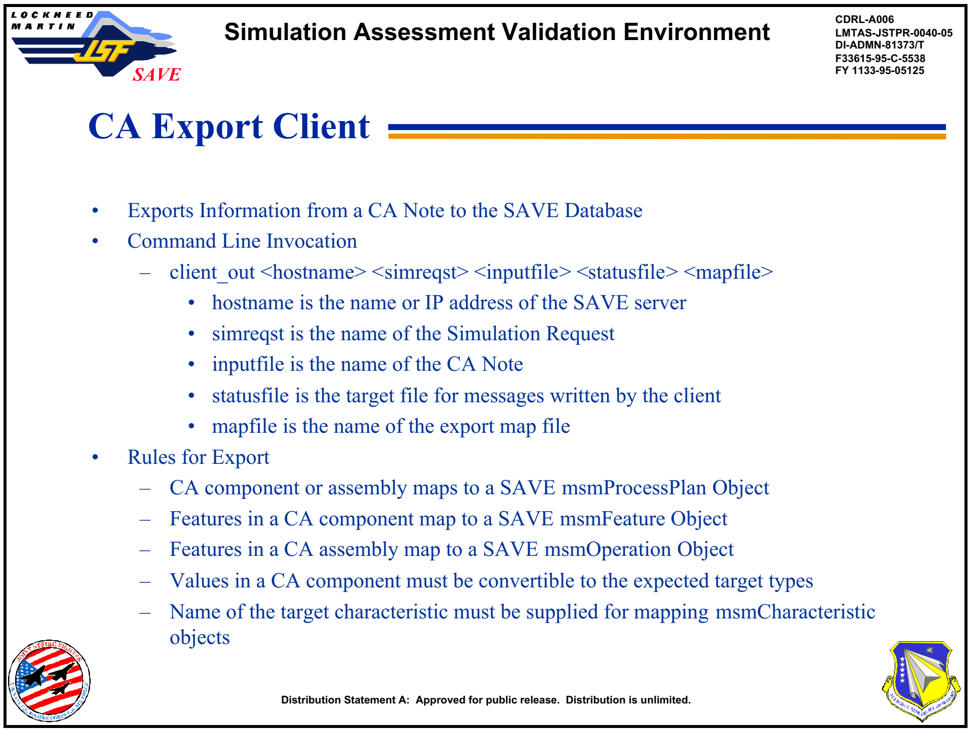
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

Mapping Examples Continued

- Characteristic Mapping
  - Entries on LHS are SAVE data, on RHS are CA data
  - General Expression: Mapping feature name
    - Imp;ProcessPlan;<process_name>;Operations;<op_name>;Name;<->;Feature;<ca_feature_type>;<ca_char_name>;
  - Specific Expression: Map SAVE operation name to a specific CA feature characteristic (Drill to ManualDrill)
    - Imp;ProcessPlan;DRILL;Operations;DRILL[0-9)*;Name;<->;Feature;ManDrill;ManDrillName;

Distribution Statement A: Approved for public release. Distribution is unlimited.



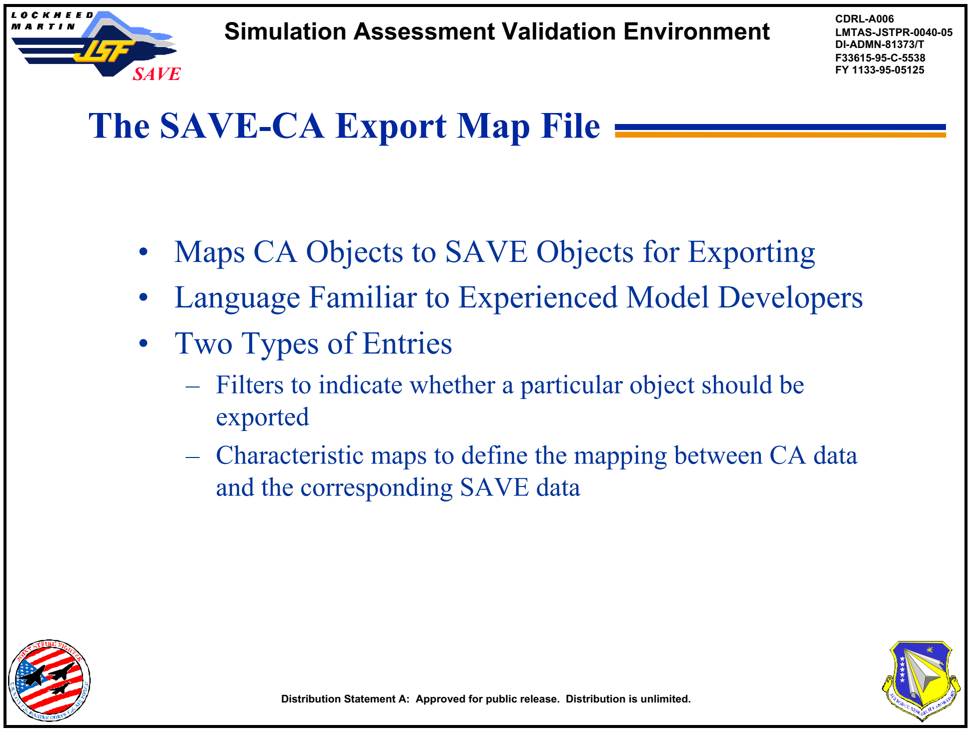
Simulation Assessment Validation Environment

CDRL-A006
LMTAS-JSTPR-0040-05
DI-ADMN-81373/T
F33615-95-C-5538
FY 1133-95-05125

CA Export Client

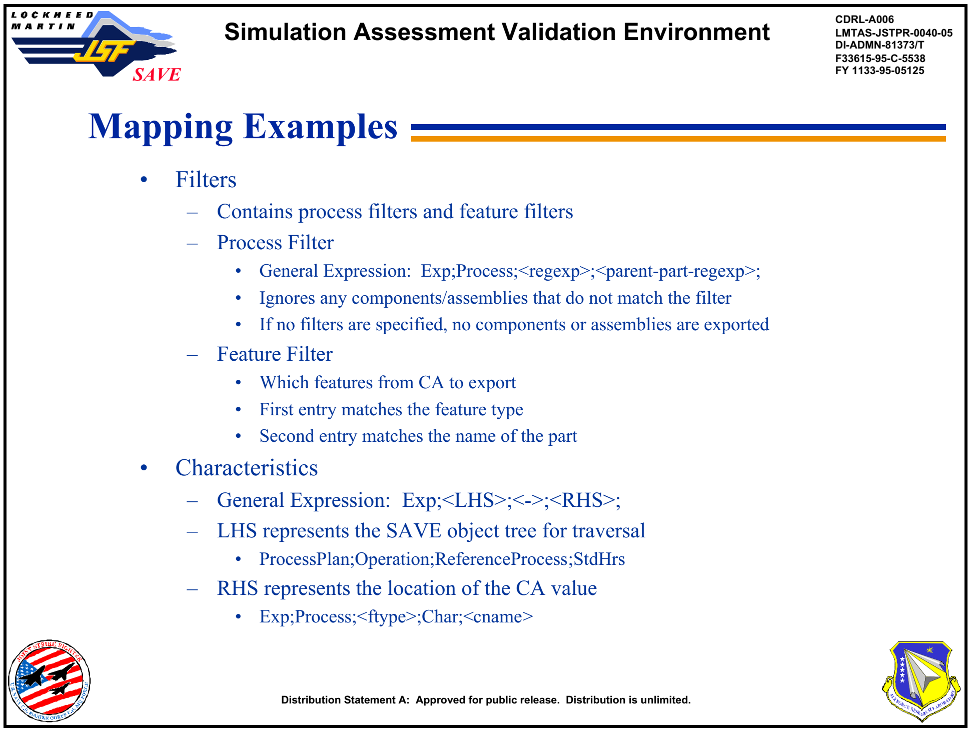
- Exports Information from a CA Note to the SAVE Database
- Command Line Invocation
  - client_out <hostname> <simreqst> <inputfile> <statusfile> <mapfile>
    - hostname is the name or IP address of the SAVE server
    - simreqst is the name of the Simulation Request
    - inputfile is the name of the CA Note
    - statusfile is the target file for messages written by the client
    - mapfile is the name of the export map file
- Rules for Export
  - CA component or assembly maps to a SAVE msmProcessPlan Object
  - Features in a CA component map to a SAVE msmFeature Object
  - Features in a CA assembly map to a SAVE msmOperation Object
  - Values in a CA component must be convertible to the expected target types
  - Name of the target characteristic must be supplied for mapping msmCharacteristic objects

Distribution Statement A: Approved for public release. Distribution is unlimited.



The SAVE-CA Export Map File

- Maps CA Objects to SAVE Objects for Exporting
- Language Familiar to Experienced Model Developers
- Two Types of Entries
  - Filters to indicate whether a particular object should be exported
  - Characteristic maps to define the mapping between CA data and the corresponding SAVE data



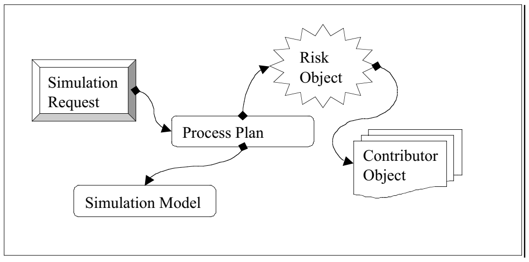
Mapping Examples

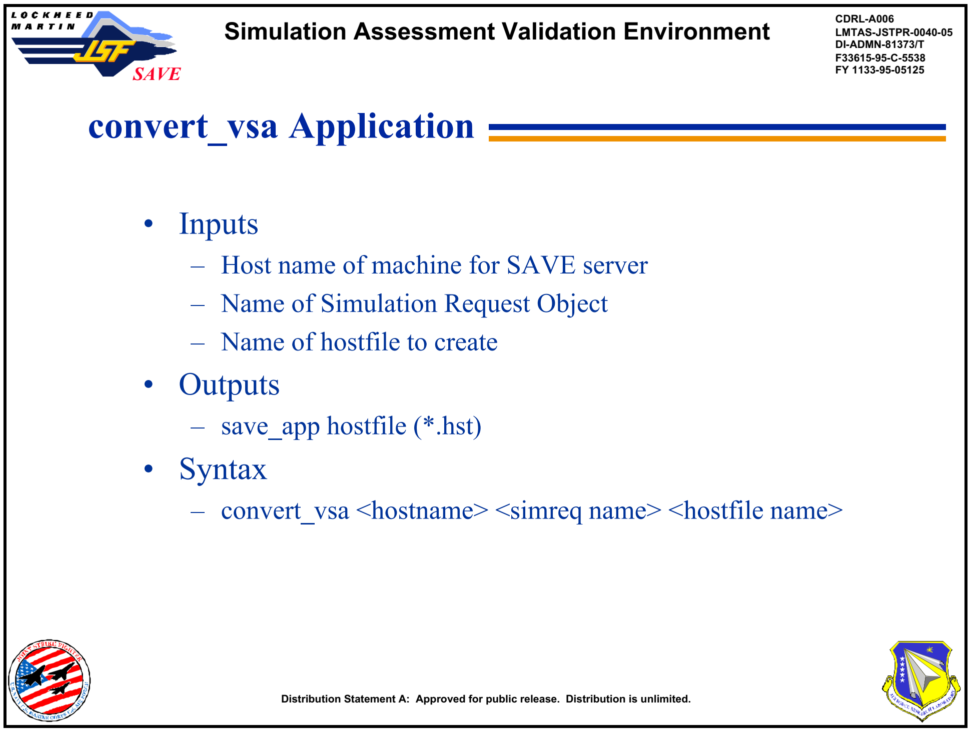
- Filters
  - Contains process filters and feature filters
  - Process Filter
    - General Expression: Exp;Process;<regexp>;<parent-part-regexp>;
    - Ignores any components/assemblies that do not match the filter
    - If no filters are specified, no components or assemblies are exported
  - Feature Filter
    - Which features from CA to export
    - First entry matches the feature type
    - Second entry matches the name of the part
- Characteristics
  - General Expression: Exp;<LHS>;<->;<RHS>;
  - LHS represents the SAVE object tree for traversal
    - ProcessPlan;Operation;ReferenceProcess;StdHrs
  - RHS represents the location of the CA value
    - Exp;Process;<ftype>;Char;<cname>

VSA 3D Wrapper Overview

VSA-SAVE Interfaces

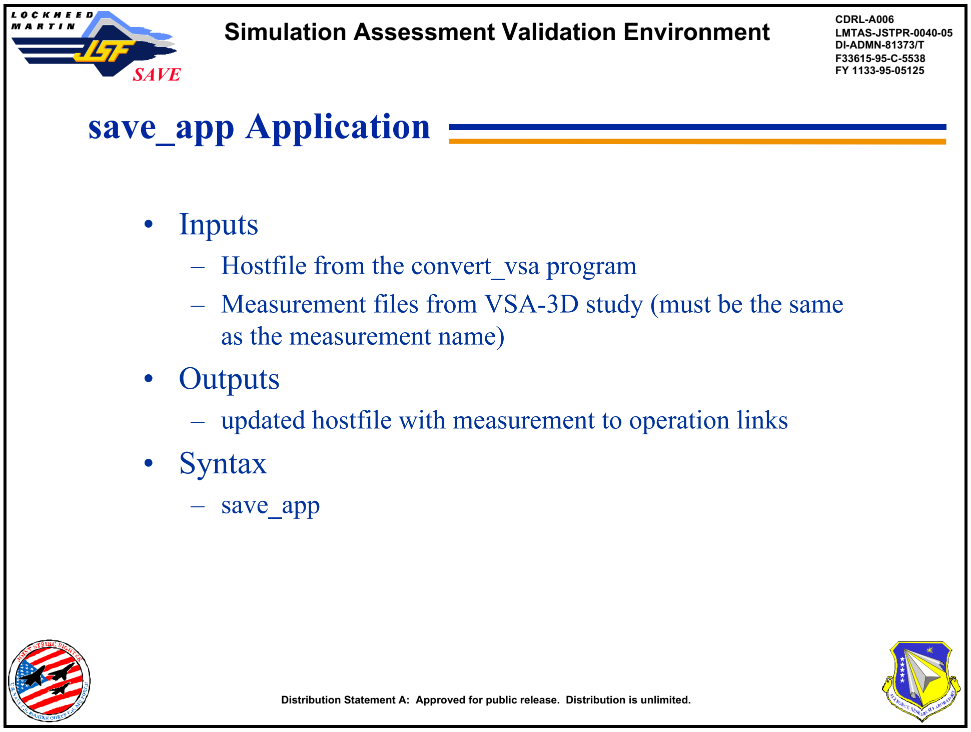
- **convert_vsa**: Extracts SAVE Process Plan into ASCII File
- **save_app**: GUI to Associate VSA Measurements with SAVE Operations
- **pop_vsa**: Extracts VSA-3D Results and Stores in SAVE





convert_vsa Application

- Inputs
  - Host name of machine for SAVE server
  - Name of Simulation Request Object
  - Name of hostfile to create
- Outputs
  - save_app hostfile (*.hst)
- Syntax
  - convert_vsa <hostname> <simreq name> <hostfile name>



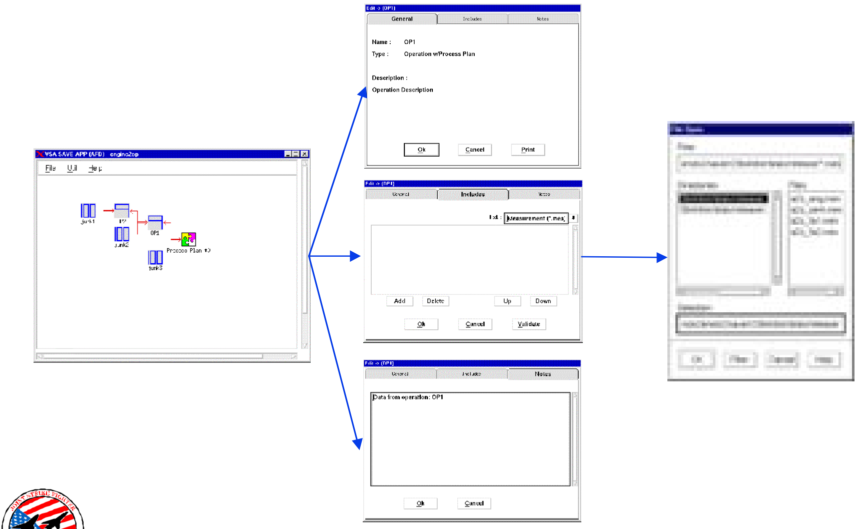
save_app Application

- Inputs
  - Hostfile from the convert_vsa program
  - Measurement files from VSA-3D study (must be the same as the measurement name)
- Outputs
  - updated hostfile with measurement to operation links
- Syntax
  - save_app

save_app Interface

1. Edit working directory using UTILS-Preferences-Current to the location of the VSA-3D model files and hostfile
2. Select hostfile using File-Open menu
3. Double click on any icon to access its properties
4. Property panel includes three tabs
   - General
   - Includes
   - Notes
5. Use Includes panel to associate measurements with operations

save_app Interface

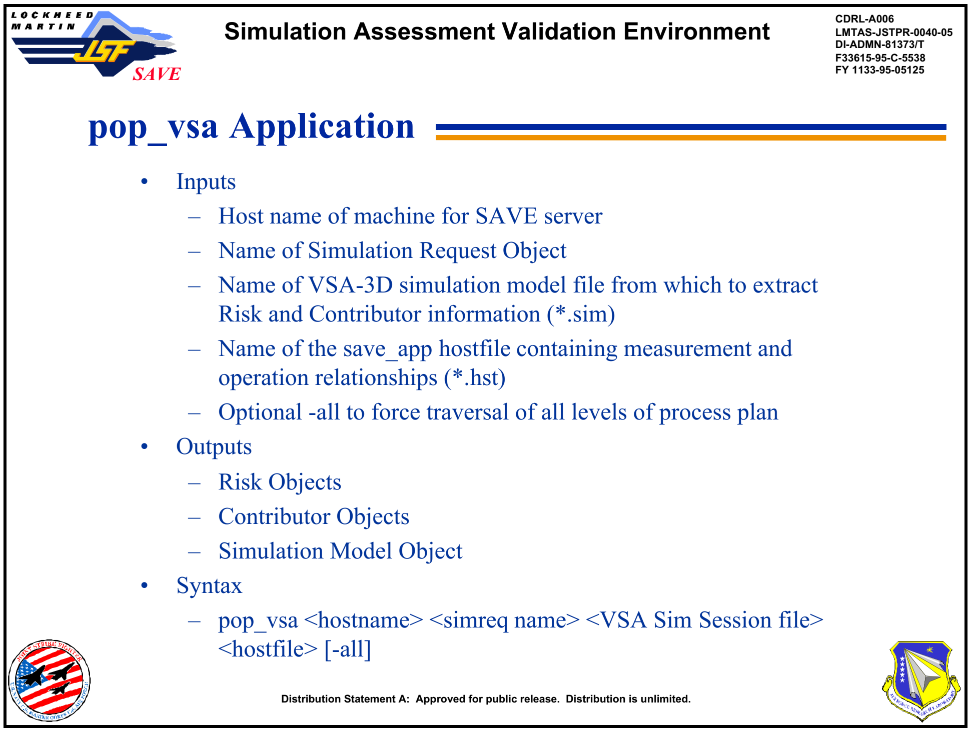


Name : OP1
Type : Operation w/Process Plan

Description :
Operation Description

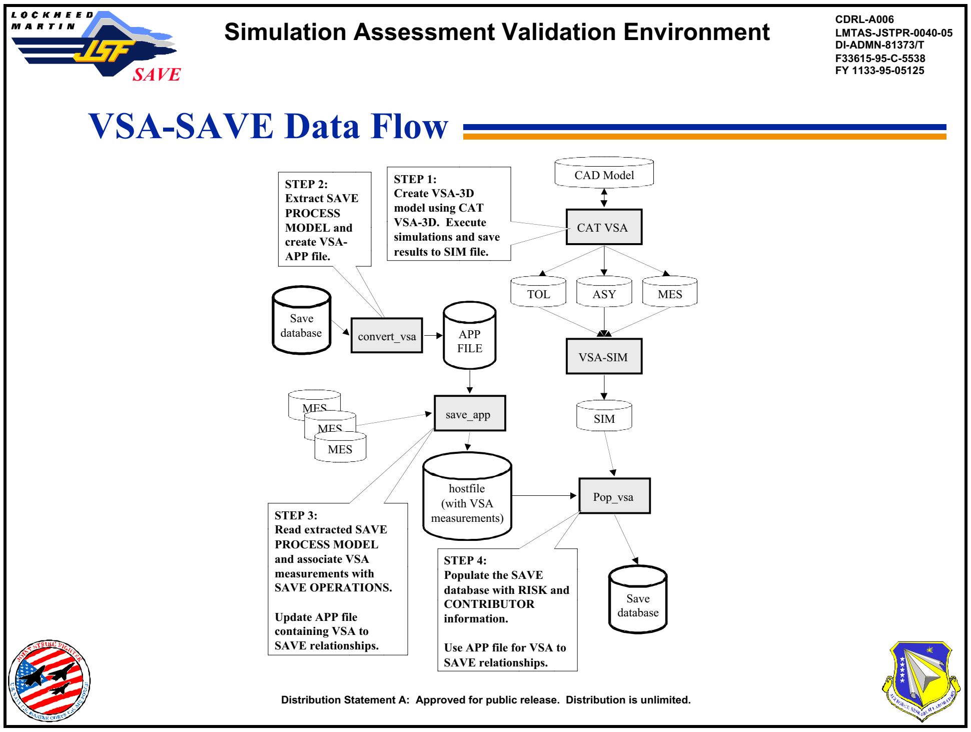
Measurement (*.mes)

Data from operation: OP1



pop_vsa Application

- Inputs
  - Host name of machine for SAVE server
  - Name of Simulation Request Object
  - Name of VSA-3D simulation model file from which to extract Risk and Contributor information (*.sim)
  - Name of the save_app hostfile containing measurement and operation relationships (*.hst)
  - Optional -all to force traversal of all levels of process plan
- Outputs
  - Risk Objects
  - Contributor Objects
  - Simulation Model Object
- Syntax
  - pop_vsa <hostname> <simreq name> <VSA Sim Session file> <hostfile> [-all]



VSA-SAVE Data Flow

STEP 1:
Create VSA-3D model using CAT VSA-3D. Execute simulations and save results to SIM file.

STEP 2:
Extract SAVE PROCESS MODEL and create VSA-APP file.

STEP 3:
Read extracted SAVE PROCESS MODEL and associate VSA measurements with SAVE OPERATIONS.

Update APP file containing VSA to SAVE relationships.

STEP 4:
Populate the SAVE database with RISK and CONTRIBUTOR information.

Use APP file for VSA to SAVE relationships.

APPENDIX O

Tool Specific Input/Output Capabilities

SAVE Software User's Manual
Contract Number F33615-95-C-5538
CDRL A012

The table shown below identifies how the simulation tools currently integrated into SAVE utilize the Process Plan within the SAVE Data Model. The fully expanded nesting of data within a Process Plan is shown, and each data field notes which tools interact with that field as Input (I), Output(O), or both (I/O). This table is current as of the SAVE Final Demonstration, completed in July 1999.

This matrix is an excellent planning aid for developing new tool interfaces, as it clearly shows which data can be provided by other tools, and what data other tools expect to be able to use as input.

.

Class, Attribute Name	ASURE	Cost Advantage	Factor Aim	IGRIP	Quest	VSA
Process Plan						
Average Time Critical Path		I/O				
Date Time						I
Description	I/O		I/O	I/O	I/O	I/O
Name		I/O	I/O	I/O	I/O	I/O
Status	I	I	I	I	I	I
Waiting Time		I/O	O		I	
Process Plan/Characteristics						
Date Time						I
Description						I
Name		I/O				I
Numerical Value						I
Textual Value		I/O				I
Process Plan/Cost						
Average Production Unit Cost		O				
Base Year		I				
Development Tooling Cost		O				
Fiscal Year		O				
Labor Inflation Factor		I				
Material Inflation Factor		I				
Material Cost		O				
Non-Recurring Tooling Cost		O				
Other Non-Recurring Manufacturing Cost		O				
Other Recurring Manufacturing Cost		O				
Plant Equipment Cost		O				
Production Tooling Cost		O				
Quality Assurance Cost		O				
Recurring Manufacturing Labor Cost		O				
Sustaining Tooling Cost		O				
Tool Material Cost		O				

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Sustaining Tooling Cost		O				
Tool Material Cost		O				
Total Manufacturing Cost		O				
Type		O				
Inflation Table						
Process Plan/Manufacturing Order						
Date Time						
Description						
Name						
Quantity						
Process Plan/Operation						
Critical Path Step		I/O				
Date Time						
Description			I/O	I/O	I/O	I
ID		I/O	I/O	I/O	I/O	I
Name	I/O	I/O	I/O	I/O	I/O	I
Precedent Operations	I/O		I/O		I/O	I
Quantity	I/O	I/O		I/O	I/O	I
Queue Avg Capacity						
Queue Duration						
Queue Total Capacity						
Run Time		I/O	I/O	O	I/O	
Setup Description						
Setup Duration		I		I/O	I/O	
Type		I/O				I
Process Plan/Operation/ Characteristics						
Date Time		I/O				I
Description						I
Name	I/O	I/O				I
Textual Value	I/O	I/O				I
Numerical Value	I/O					I
Process Plan/Operation/Cost						
Average Production Unit Cost		O				
Base Year		I				
Development Tooling Cost		O				
Fiscal Year		O				

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Labor Inflation Factor		I				
Material Inflation Factor		I				
Material Cost		O				
Non-Recurring Tooling Cost		O				
Other Non-Recurring Manufacturing Cost		O				
Other Recurring Manufacturing Cost		O				
Plant Equipment Cost		O				
Production Tooling Cost		O				
Quality Assurance Cost		O				
Recurring Manufacturing Labor Cost		O				
Sustaining Tooling Cost		O				
Tool Material Cost		O				
Total Manufacturing Cost		O				
Type		O				
Inflation Table						
Process Plan/Operation/ Features						
Date Time						
Description						I
Name	I/O					I
Quantity	I/O					I
Type						I
Process Plan/Operation/Features/ Characteristics						
Date Time						
Description						
Name	I/O					
Textual Value	I/O					
Numerical Value	I/O					
Process Plan/Operation/Features/Cost						
Average Production Unit Cost						
Base Year						
Development Tooling Cost						
Fiscal Year						
Labor Inflation Factor						

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Material Inflation Factor						
Material Cost						
Non-Recurring Tooling Cost						
Other Non-Recurring Manufacturing Cost						
Other Recurring Manufacturing Cost						
Plant Equipment Cost						
Production Tooling Cost						
Quality Assurance Cost						
Recurring Manufacturing Labor Cost						
Sustaining Tooling Cost						
Tool Material Cost						
Total Manufacturing Cost						
Type						
Inflation Table						
Process Plan/Operation/Part						
Associated Parts				O		
Complexity						
Date Time						
Description				I/O	I/O	
Family				I/O	I/O	
Name	I/O			I/O	I/O	
Number				I/O		
Quantity	I/O				I/O	
Rejection Rate	I/O					
Type						
Process Plan/Operation/Part/CAD Model						
Date Time						
Description				I/O	I/O	
Envelope X, Y, Z						
Format				O	O	
Location				I/O	I/O	
Name				I/O		
Process Plan/Operation/Part/Cost						
Average Production Unit Cost						

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Base Year						
Development Tooling Cost						
Fiscal Year						
Labor Inflation Factor						
Material Inflation Factor						
Material Cost						
Non-Recurring Tooling Cost						
Other Non-Recurring Manufacturing Cost						
Other Recurring Manufacturing Cost						
Plant Equipment Cost						
Production Tooling Cost						
Quality Assurance Cost						
Recurring Manufacturing Labor Cost						
Sustaining Tooling Cost						
Tool Material Cost						
Total Manufacturing Cost						
Type						
Inflation Table						
Process Plan/Operation/Part/Feature						
Date Time						
Description						
Name	I/O					
Quantity	I/O					
Type						
Process Plan/Operation/Part/Feature/ Characteristics						
Date Time						
Description						
Name	I/O					
Textual Value	I/O					
Numerical Value	I/O					
Process Plan/Operation/Part/Feature/Cost						
Average Production Unit Cost						
Base Year						

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Development Tooling Cost						
Fiscal Year						
Labor Inflation Factor						
Material Inflation Factor						
Material Cost						
Non-Recurring Tooling Cost						
Other Non-Recurring Manufacturing Cost						
Other Recurring Manufacturing Cost						
Plant Equipment Cost						
Production Tooling Cost						
Quality Assurance Cost						
Recurring Manufacturing Labor Cost						
Sustaining Tooling Cost						
Tool Material Cost						
Total Manufacturing Cost						
Type						
Inflation Table						
Process Plan/Operation/Part/Material						
Date Time						
Description						
Form						
Name						
Type						
Unit Cost						
Process Plan/Operation/Part/Material/ Characteristics						
Date Time						
Description						
Name						
Textual Value						
Numerical Value						
Process Plan/Operation/ Process Plan Refer Back to Top of Page						

Class, Attribute Name	ASURE	Cost Advantage	Factor Aim	IGRIP	Quest	VSA
Process Plan/Operation/ Reference Process						
Complexity	I/O					
Date Time						
Description						
Maturity	I/O					
Name	I/O					
Operation Characteristics						
Stability						
Standard Hours						
Process Plan/Operation/ Reference Process/ Characteristics						
Date Time						
Description						
Name	I/O					
Textual Value	I/O					
Numerical Value	I/O					
Process Plan/Operation/ Reference Process/Risk						
Consequence of Failure						
Cp						I/O
Cpk						I/O
Description						I/O
Mean	I/O					I/O
Probability of Failure	O					I/O
Qualitative Results						I/O
Standard Deviation						I/O
Yield	O					I/O
Process Plan/Operation/ Reference Process/Risk/ Contributors						
Date Time						I/O
Description						I/O
Name	I/O					I/O
Percent Contribution	I/O					I/O
Process Plan/Operation/ Resource Application						
Date Time						
Description						
Name						
Transformation Matrix			I/O	I/O	I/O	
Quantity					I/O	
Process Plan/Operation/ Resource Application/ Resource						
Date Time						
Description			I/O	I/O	I/O	
Name			I/O	I/O	I/O	
Efficiency						

Class, Attribute Name	ASURE	Cost Advantage	Factor Aim	IGRIP	Quest	VSA
Process Plan/Operation/ Resource Application/ Resource/CAD Model						
Date Time						
Description						
Envelope X, Y, Z						
Format				O	O	
Location				I/O	I/O	
Name						
Process Plan/Operation/ Resource Application/ Resource Pool						
Date Time						
Description			I/O	I/O	I/O	
Name			I/O	I/O	I/O	
Quantity			I/O	I/O	I/O	
Utilization					O	
Process Plan/Operation/ Resource Application/ Resource Pool/Resource						
Date Time						
Description		I/O	I/O	I/O	I/O	
Name			I/O	I/O	I/O	
Efficiency		I		I/O		
Process Plan/Operation/ Resource Application/ Resource Pool/Resource/CAD Model						
Date Time						
Description						
Envelope X, Y, Z						
Format				O	O	
Location				I/O	I/O	
Name						
Process Plan/Operation/Risk						
Consequence of Failure						I/O
Cp						I/O
Cpk						I/O
Description						I/O
Mean	I/O					I/O
Probability of Failure	O					I/O
Qualitative Results						I/O
Standard Deviation						I/O
Yield	O					I/O
Process Plan/Operation/Risk/Contributors						
Date Time						I/O
Description						I/O
Name	I/O					I/O
Percent Contribution	I/O					I/O

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Process Plan/Operation/ Schedule						
Actual Duration				O	O	
Actual End Date						
Actual Start Date						
Planned Duration						
Planned End Date						
Planned Start Date						
Priority						
Process Plan/Part						
Associated Parts						
Complexity						
Date Time						
Description						
Family		O				
Name	I/O	I/O				
Number		O				
Quantity	I/O					
Rejection Rate	I/O					
Type		I/O				
Process Plan/Part/ CAD Model						
Date Time						
Description						
Envelope X, Y, Z						
Format						
Location						
Name						
Process Plan/Part/ Cost						
Average Production Unit Cost		O				
Base Year		I				
Development Tooling Cost		O				
Fiscal Year		O				
Labor Inflation Factor		I				
Material Inflation Factor		I				
Material Cost		O				
Non-Recurring Tooling Cost		O				
Other Non-Recurring Manufacturing Cost		O				
Other Recurring Manufacturing Cost		O				
Plant Equipment Cost		O				

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Production Tooling Cost		O				
Quality Assurance Cost		O				
Recurring Manufacturing Labor Cost		O				
Sustaining Tooling Cost		O				
Tool Material Cost		O				
Total Manufacturing Cost		O				
Type		O				
Inflation Table						
Process Plan/Part/ Feature						
Date Time						
Description						
Name	I/O					
Quantity	I/O					
Type						
Process Plan/Part/ Feature/ Characteristics						
Date Time						
Description						
Name	I/O					
Textual Value	I/O					
Numerical Value	I/O					
Process Plan/Part/ Feature/Cost						
Average Production Unit Cost						
Base Year						
Development Tooling Cost						
Fiscal Year						
Labor Inflation Factor						
Material Inflation Factor						
Material Cost						
Non-Recurring Tooling Cost						
Other Non-Recurring Manufacturing Cost						
Other Recurring Manufacturing Cost						
Plant Equipment Cost						
Production Tooling Cost						

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Quality Assurance Cost						
Recurring Manufacturing Labor Cost						
Sustaining Tooling Cost						
Tool Material Cost						
Total Manufacturing Cost						
Type						
Inflation Table						
Process Plan/Part/ Material						
Date Time						
Description						
Form						
Name		I				
Type		O				
Unit Cost						
Process Plan/Part/ Material/ Characteristics						
Date Time						
Description						
Name						
Textual Value						
Numerical Value						
Process Plan/Resource Pool						
Date Time						
Description			I/O	I/O	I/O	
Name			I/O	I/O	I/O	
Quantity			I/O	I/O	I/O	
Utilization					O	
Process Plan/Resource Pool/ Resource						
Date Time						
Description			I/O	I/O	I/O	
Name			I/O	I/O	I/O	
Efficiency		I/O				
Process Plan/Resource Pool/ Resource/CAD Model						
Date Time						
Description						
Envelope X, Y, Z						
Format				O	O	
Location				I/O	I/O	
Name						

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Process Plan/Risk						
Consequence of Failure						
Cp						
Cpk						
Description						
Mean	I/O					
Probability of Failure	O					
Qualitative Results						
Standard Deviation						
Yield	O					
Process Plan/Risk/ Contributors						
Date Time						
Description						
Name	I/O					
Percent Contribution	I/O					
Process Plan/Schedule						
Actual Duration			O			
Actual End Date			O			
Actual Start Date			O			
Planned Duration						
Planned End Date						
Planned Start Date						
Priority						
Process Plan/Simulation Model						
Data Location				O	O	O
Date Time				O	O	O
Description				O	O	O
Factory Model						
Name				O	O	O
Simulation Code				O	O	O
Type				O	O	O

Resource Subclass Relationship

Applies to Each Instance of Resource Unless Otherwise Noted.

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Resource						
Date Time						
Description			I/O	I/O	I/O	
Name			I/O	I/O	I/O	
Efficiency						
Resource/Personnel						
Skill				I/O		
Labor Rate						
Labor Rate Year						
Resource/Personnel/ Work Calendar						
Date Time						
Description						
Name					I/O	
Number of Work Days					I/O	
Work Year						
January 1 Day of Week						
Resource/Personnel/Work Shift						
Date Time						
Description						
End Time					I/O	
Hours in Shift						
Name					I/O	
Start Time					I/O	
Resource/Personnel/Work Shift/Break						
Start Time					I/O	
End Time					I/O	
Resource/Tools						
Cost						
Failure Rate						
Tolerance Capability						
Type						
Breakdown						
Characteristics						

Class, Attribute Name	**ASURE**	**Cost Advantage**	**Factor Aim**	**IGRIP**	**Quest**	**VSA**
Resource/Tools/Breakdown						
Date Time						
Description						
Name						
Repair Time						
Time Between Failures						
Time to First Failure						
Resource/Tools/Breakdown/Resource *See Resource Information*						
Resource/Tools/Characteristics						
Date Time						
Description						
Name						
Numerical Value						
Text Value						

Notes:

- Object nesting is depicted with color-coding as well as with specification of the full object name, including all parent objects.
- Attributes are in the list below the interface name and are shown in alphabetical order.
- Methods are not identified in this matrix.
- I/O identifies that the information can be either input into the tool or output from the tool. In some cases I/O may refer to editing capability from the tool's interface, but not its use within the simulation tool itself.
- I identifies that the information can only be input into the tool.
- O identifies that the information is only a tool output.